**PROJECT:** How to Build an AC Tester
**LOCATION:** United States
**PAGE:** 14

**INTERVIEW:** Industrial Control Engineering
**LOCATION:** Canada
**PAGE:** 24

**DIY:** Kinect-Based Image Processing
**LOCATION:** Spain
**PAGE:** 36

# CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

JUNE 2012
ISSUE 263

## COMMUNICATIONS

## Smart Switch Communication & Control System

## Radio Frequency Mixers

## Diode ORing: Maintain Uninterrupted Power

## MOSFET Channel Resistance Explained

## MCU-Based Automatic Blood Pressure Cuff

7 25274 75349 9 0 6>
$7.50 U.S. ($8.50 Canada)
www.circuitcellar.com

**PLUS**

**Concurrency 101**

Concurrency in Embedded Systems
// Common Concurrency Pitfalls
// Preventing Priority Inversion
// Managing Requirements
// And More

# TASK MANAGER

## H2M & M2M Communication

Before I introduce this issue, I'd like to bring your attention to our recently redesigned website, CircuitCellar.com. It enables engineers and programmers around the world to communicate and share ideas via project articles, videos, and social media. The site's features range from project posts (how-to articles, videos, and photos) to daily updates about products and industry news. We also run short write-ups on actual circuit cellars and workbenches in the well-received "Reader Workspaces" section of the site. I encourage you to submit photos and info about your workspaces. Share your space with the design community!

Now let's focus on this issue, which has articles on both human-to-machine (H2M) and machine-to-machine (M2M) communication. Topics as diverse as smart switch management and human motion-sensing systems are covered.

Kevin Gorga kicks off the issue with his "AC Tester" project (p. 14). It is an isolated variable voltage power source that includes an electronic circuit breaker for testing and debugging equipment. An mbed controller displays voltage and current, and it controls the breaker's trip point and response time.

*Circuit Cellar* published 15 of Aubrey Kagan's articles from 2000 to 2010. In an interview on page 24, Aubrey shares some of his engineering experiences from designing controllers for mines in Africa to helping create specs for the remote control arm on the International Space Station.

On page 28, Fergus Dixon presents a 'Net-connected controller for up to 50 smart switches for lighting systems. The controller's RTC pulses a 24-V AC line once or twice to turn off the smart switches at the end of the day.

Turn to page 36 if you're interested in computer vision technology. Miguel Sánchez introduces depth camera technology, and describes how he used Microsoft's Kinect in an interactive art project.

On page 44, columnist Bob Japenga starts an articles series on the subject of concurrency in embedded systems. In this article, he defines concurrency and covers some concurrency-related pitfalls.

Last month columnist George Novacek examined the topic of product testing and simulation. In this issue, he tackles a different yet equally important topic: diode ORing (p. 48).

On page 52, columnist Ed Nisley carefully explains MOSFET channel resistance. He describes power MOSFET operation and explores the challenges of using a MOSFET's drain-to-source resistance as a current-sensing resistor.

A serious RF designer should have a sound understanding of frequency mixers. On page 58, columnist Robert Lacoste summarizes the basics of RF mixers and presents real-life applications.

Jeff Bachiochi wraps up the issue with an article about a DIY, MCU-based blood pressure cuff project (p. 68). He converted a manual blood pressure cuff into an automatic cuff by adding an air pump, a solenoid release valve, and a pressure sensor.

cj@circuitcellar.com

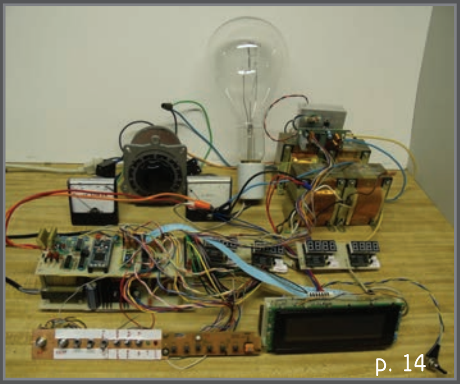
C. Abate

# INSIDE ISSUE 263

June 2012 • Communications

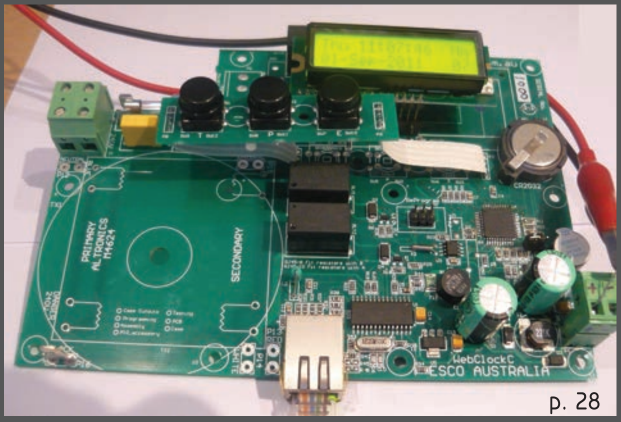


Equipment Testing and Debugging



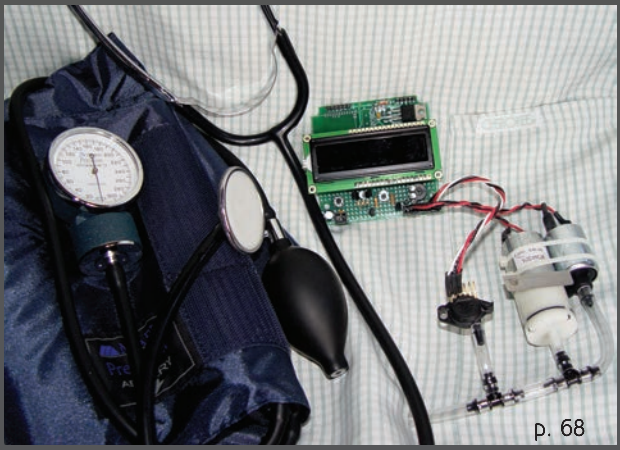
p. 14

Internet-Enabled Switch Controller



p. 28

MCU-Based Blood Pressure Cuff



p. 68


AS9120A
Certified Distributor
Scan Here
mouser.com
Semiconductors and electronic components for design engineers.
Authorized Distributor
We deliver ASAP, PDQ and JIT. So, you're never SOL.
Mouser delivers the components you need, on-time. And with local Technical Support and Customer Service Experts in 19 locations around the world, you'll find the newest components to launch your new design seamlessly.
mouser.com
The Newest Products for Your Newest Designs®
MOUSER ELECTRONICS
a tti company
Mouser and Mouser Electronics are registered trademarks of Mouser Electronics, Inc. The Newest Products for Your Newest Designs is a registered trademark of Mouser Electronics, Inc.

## THE TEAM



## THE NETWORK

audioXpress

CIRCUIT CELLAR

THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING INFORMATION

elektor

techthefuture

Tech the Future explores the solutions for a sustainable future provided by technology, creativity and science.

VOICE COIL

## OUR INTERNATIONAL TEAMS




Issue 263 June 2012 ISSN 1528-0608

CIRCUIT CELLAR® (ISSN 1528-0608) is published monthly by Circuit Cellar Incorporated, 4 Park Street, Vernon, CT 06066. Periodical rates paid at Vernon, CT and additional offices. One-year (12 issues) subscription rate USA and possessions $50, Canada $65, Foreign/ROW $75. All subscription orders payable in U.S. funds only via Visa, MasterCard, international postal money order, or check drawn on U.S. bank.

Cover photography by Chris Rakoczy—www.rakoczyphoto.com

**Subscriptions**
Circuit Cellar, P.O. Box 462256, Escondido, CA 92046
E-mail: circuitcellar@pcspublink.com
Phone: 800.269.6301, Internet: www.circuitcellar.com
Address Changes/Problems: circuitcellar@pcspublink.com

Postmaster: Send address changes to Circuit Cellar, P.O. Box 462256, Escondido, CA 92046.

**US Advertising**
Strategic Media Marketing, Inc.
2 Main Street, Gloucester, MA 01930 USA
Phone: 978.281.7708, Fax: 978.281.7706, E-mail: peter@smmarketing.us
Internet: www.circuitcellar.com
Advertising rates and terms available on request.

New Products: New Products, Circuit Cellar, 4 Park Street, Vernon, CT 06066, E-mail: newproducts@circuitcellar.com

## Membership counter

We now have 277285 members in 83 countries.

**Not a member yet?**

Sign up at www.circuitcellar.com

## Supporting companies

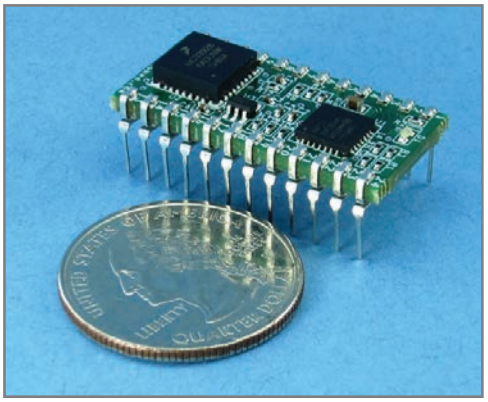


**Not a supporting company yet?**

Contact Peter Wostrel (peter@smmarketing.us, Phone 978.281.7708, Fax 978.281.7706) to reserve your own space for the next issue of our member's magazine.


**Head Office**
Circuit Cellar, Inc.
4 Park Street, Vernon, CT 06066, Phone: 860.875.2199

## MINIATURIZED MOTION CONTROLLERS



The **Synaptron Micro** is a tiny, 24-pin, dual in-line package (DIP), miniaturized motion controller. It controls DC motors powered by 6-to-24 VDC and can handle a current load of 1.5-A continuous (5-A peak). The module is well suited for open-loop motor pulse-width modulation (PWM) drive controls or closed-loop motion control.

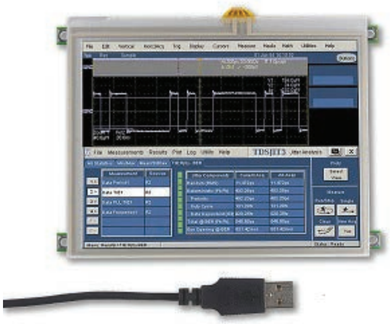
The Synaptron Micro features a host of user settings that enable it to accept control inputs from serial commands, pulses, or analog signals. The module also features configurable feedback options including: analog signals, pulse or frequency signals, quadrature encoders, and internal current or velocity calculations.

The Synaptron Micro includes free test software to simplify user customization of control and feedback signals, proportional integral derivative (PID) programming, limit-switch settings (including virtual limit switches), PWM limits, current limits, and many other adjustments.

The modules cost **$50** per unit and **$40** in 250-unit quantities.

**Solutions Cubed, LLC**
**www.solutions-cubed.com**

## SMART TOUCH LCD FOR 5.6" EMBEDDED DISPLAY APPLICATIONS



The **ezLCD-005** graphic display module features an all-in-one design that combines a color LCD, a touchscreen, control electronics, memory and I/O with an easy-to-use, command-driven programmable firmware environment. The module is designed to help engineers minimize development time and reduce hardware costs, while speeding time to market for applications requiring a color touchscreen display.

Features of the ezLCD-005 include: 5.6" color TFT LCD with LED backlight; an integrated four-wire resistive touchscreen; 320 x 234 resolution supporting 65,536 colors; USB, SPI, and RS-232/TTL interfaces; 0 to 60°C operating temperature; 1-MB serial flash memory; an SD/MMC card slot for memory expansion up to 4 GB; and compatibility with any microcontroller.

Pricing for the ezLCD-005 starts at **$330**.

**Earth Computer Technologies, Inc.**
**http://store.earthlcd.com**

## WIRELESS SOLUTIONS FOR NIAGARA FRAMEWORK

SCL Elements—developer of CAN2GO controllers—and Tridium—developer of the Niagara Framework—announced the integration capability of EnOcean and ZigBee technologies into Niagara-based building automation systems. The new capability enables the Niagara community to broaden its use of wireless solutions and reduce wiring and repair costs for energy-efficiency commissioning and retrofit deployments.

The integration of EnOcean- and ZigBee-based products to Niagara-based JACEs is accomplished with CAN2GO controllers, which support bidirectional EnOcean and ZigBee communication and push the wireless points to JACE hardware via BACnet or Obix. The integration includes: HVAC, lighting, occupancy, metering, and other building applications.

**SCL Elements, Inc.**
**www.can2go.com**

**Tridium, Inc.**
**www.tridium.com**

NEW PRODUCT NEWS


Imagine,
Design,
EARN!
club JAMECO®

by Ian Bromley

# DesignSpark chipKIT™ Design Challenge Community Choice Winners

**The DesignSpark chipKIT™ design challenge for energy-efficient applications is now closed. Here is a quick look at a few of the Community Choice Award Projects.**

The response to the DesignSpark chipKIT™ competition, which closed for final entries at the end of March, has been overwhelming. What's more is that we've been really encouraged with the high quality of the submitted designs. Just in case any readers are hearing about this for the first time, the aim of the DesignSpark chipKIT™ challenge has been to encourage engineers, students, and hobbyists to develop new and innovative energy-efficient solutions, while also maintaining an eco-friendly footprint.

The competition is based on the chipKIT™ Max32™ Arduino-compatible development platform from Digilent, which features Microchip Technology's 32-bit PIC32 microcontroller and enables developers to easily and inexpensively integrate electronics into their projects. All entries had to include an extension card developed using RS Components's free-of-charge and award-winning DesignSpark PCB software tool.

We're now busy evaluating all the entries to find the winning designs. The winners will be announced in May and will receive total cash prizes of $10,000, including a first prize of $5,000. However, along the way we have also been announcing DesignSpark Community Choice award winners and runners-up. During the competition, entrants were strongly encouraged to engage and interact with other members of the online DesignSpark community, available at www.designspark.com, by posting information about their projects, providing updates on their progress, and sharing comments and ideas on their respective designs.

The Community Choice award winner for January was Brian Connell from South Africa. I've mentioned his project before in a previous column: it is a highly useful design for implementation in parts of the world where water is in short supply. The project is a hydroponics water and nutrient-control system that will monitor climatic conditions to determine the level of nutrient-feed requirements for plants in a hydroponic environment, thereby saving on water usage and nutrient chemicals. Most hydroponics farms currently use established feed regimes that run without adjustment for environmental conditions, so the design could deliver both financial and water savings.

We also had a couple of highly interesting runner-up projects for January. One of them is a Lego brick-sorting device that uses image-processing technology. The device sorts bricks by color and size via the processing of an image taken by either a webcam or an RGB sensor and includes a stepper motor driven by a Texas Instruments DRV8412 evaluation kit, controlled by the chipKIT board. A second runner-up was an innovative design for a "21st century barrel organ." The six-voice polyphonic instrument uses the energy created by a musician turning a crank connected to an electro-generator to power the electronics and sound emitters.

Turning to February, the Community Choice winner was Curtis Brooks from the U.S. for an Internet-enabled multizone thermostat. The design idea was that each room in a house would have network node that connects to an Internet router, the central node, via XBee Wi-Fi modules. The localized nodes control the thermostats located in each room and will be able to automatically adjust the heating or cooling, based on preset levels, with data from each node monitored and adjusted through the Internet. The overall system aims to conserve energy by adjusting heating based on room requirements, which will be more efficient than single-zone thermostats. The design can potentially also be used to control hot water heating systems.

Anyway, to finish, I'd really like to thank everyone who has entered in the competition and all the contributors to the discussions on the projects. You have made it a fantastic success. And now, in the coming weeks, we're looking forward to announcing the overall winners and the Community Choice awards for the final month of the competition. If you're interested in reading more about the competition and want to take a closer look at all the design entries, please go to www.designspark.com/chipkitchallenge.

*Ian Bromley is a Technical Marketing Engineer at RS Components and the Project Manager for the DesignSpark PCB software tool. Prior to working for RS, Ian worked for many years as a design support consultant with Texas Instruments, in addition to working as a field applications engineer immediately following his graduation in 1994 with an honors degree in Microelectronic Engineering.*

**Check out projects from the competition at www.designspark.com/chipkitchallenge**

# Test Your EQ

Edited by David Tweed

## ANSWERS for Issue 262

**Answer 1**—Right before the moment Q1 switches on, C1 is charged to $V_{CC} - V_{BE}$, with its left end positive, and the left end of C2 has just reached $+V_{BE}$. The right end of C2 is being held at $V_{CE(SAT)}$ by Q2.

So, as Q1 begins to switch on, it pulls the left end of C1 low, and this also pulls the right end of C1 low, cutting off Q2. This in turn enables the right end of C2 to rise, emphasizing the turn-on of Q1 by increasing the voltage (and current) at the base of Q1.

Once Q1 is fully on, the right end of C1 is now at $V_{CE(SAT)} - (V_{CC} - V_{BE})$ (a fairly substantial negative voltage), and C1 now begins to charge in the other direction, through R2. Once the right end of C1 reaches $+V_{BE}$, Q2 begins to turn on, starting the second half of the cycle.

**Answer 2**—The time of the half-cycle described previously is the time that it takes the right end of C1 to charge from $-(V_{CC} - (V_{BE} + V_{CE(SAT)}))$ to $+V_{BE}$.

Now, keep in mind that the capacitor is charging "toward" $+V_{CC}$, but it gets halted by the B-E junction of Q2 at $+V_{BE}$. This charging is occurring at a rate determined by the time constant C1 × R2, and we're basically interested in the time that it takes to move halfway from its starting value to its final value. This works out to $-\ln(0.5)$, or 0.693 times the R-C time constant.

As long as $V_{CC} >> V_{BE}$, the time does not depend on $V_{CC}$. That isn't to say, however, that $V_{CC}$ can be arbitrarily large. If it exceeds the reverse-breakdown voltage of the transistors' B-E junctions, current will flow and perturb the timing.

**Answer 3**—No, it isn't. This circuit can oscillate just fine. Again, look at how C1 charges and discharges.

If C1 starts out discharged, it will charge through R1 and the B-E junction of Q2. This current will turn on Q2, holding its collector at ground (really $V_{CE(SAT)}$) and preventing Q1 from turning on.

However, as C1 reaches full charge, the current through it decays below a level that will keep Q2 turned on. When it starts to turn off, its collector voltage rises, which also forces current into Q1's base through C2. As Q1 begins to turn on, it pulls its collector low, which also pulls the base of Q2 lower, emphasizing its turn-off. The circuit quickly "snaps" to the other state, with Q1 on and Q2 off. C1 is discharged through Q1 and D2 at the same time that C2 begins charging through R4 and Q1's B–E junction.

**Answer 4**—R2 and R3 never have more than $\pm V_{BE}$ across them; as a result, the current through them is negligible relative to the current through the capacitors. In other words, they're superfluous.

**Answer 5**—The time from when one of the transistors turns on to when it turns off is determined by the currents flowing into its base and collector. When the current into the base drops below the value needed to sustain the current into the collector, the transistor begins to turn off, and the circuit feedback then ensures that this happens quickly.

Looking at Q2, and ignoring the transient associated with discharging C2 for now, the collector current is set by R4. The initial base current is set by R1, but this decays exponentially with a time constant of R1 × C1.

Therefore, the primary determinant of the half-cycle time period (in addition to the R-C time constant) is the current transfer ratio, or $h_{FE}$ of each transistor. When the base current drops to a value of $1/h_{FE}$ of the collector current, the transistor begins to turn off.

Since both currents scale in the same way with $V_{CC}$, it has no direct effect on the timing. There is only a secondary effect if the value of $h_{FE}$ changes with the value of the collector current.

*Contributed by David Tweed*

**What's your EQ?—The answers are posted at www.circuitcellar.com/eq/**

**You may contact the quizmasters at eq@circuitcellar.com**

by Kevin Gorga (USA)

# AC Tester

## Testing, Troubleshooting, and Debugging

Safety is a top priority when working with electronics and circuits. The AC Tester design is an isolated variable voltage power source that includes an electronic circuit breaker for testing and debugging equipment. An mbed controller displays voltage and current, and it controls the breaker's trip point and response time. In addition, this inventive design can display power factor, VA, and VAR.

For many years, I have had a variac transformer plugged into an isolation transformer with a lightbulb in series and a pair of alligator clips on the end of a line cord. This is my most useful piece of test equipment for repairing and testing line-operated equipment. It provides safety from shock and short circuits. The isolation transformer provides the safety for equipment that has one side of the line as a common. With the isolation transformer in place, the scope probe can be ground-clipped to one side of an offline power supply with safety. The series lightbulb provides for current limiting when a short occurs or your probe slips. The variac provides a means to slowly apply power to a piece of equipment or to verify proper operation under fluctuating line voltage.

In this article, I'll describe my NXP Semiconductors mbed-based AC Tester design (see Photo 1). I use it for prototyping, testing, and repairing equipment that is powered off the AC line (see Figure 1). The AC Tester provides power line isolation to enable the safe use of an oscilloscope on nonisolated equipment. It provides adjustable line voltage to test proper operation at all line levels. It displays on both analog (for quick recognition) and digital (for accuracy) volt and ammeters. It also has a series incandescent lightbulb to provide protection and indication of overcurrent conditions. If a repair or prototype has a high-power short, the series bulb will illuminate and limit the power to the device. A variable response time solid-state circuit breaker is also incorporated for safely cutting off power at any set current and response time. Peak surge current is also displayed. An optional energy meter (E meter) is also provided for displaying of watts, VA, and VAR. Power factor is also available.

## DESIGN OVERVIEW

The design is partitioned so that it can be built in three forms. The simplest version is the variac, isolation transformer, and analog meters. The next level up incorporates the microcontroller to enable a more accurate digital voltage and current display. It also adds the variable response time electronic circuit breaker. The ultimate version adds an electronic E meter for watt, VA, and VAR display.

I used the I²C, analog-to-digital (A/D), digital-to-analog (D/A), SPI, and interrupts functions of an NXP Semiconductors LPC1768 microcontroller. I²C is used to expand the I/O capabilities. Four NXP Semiconductors SAA1064 LED drivers

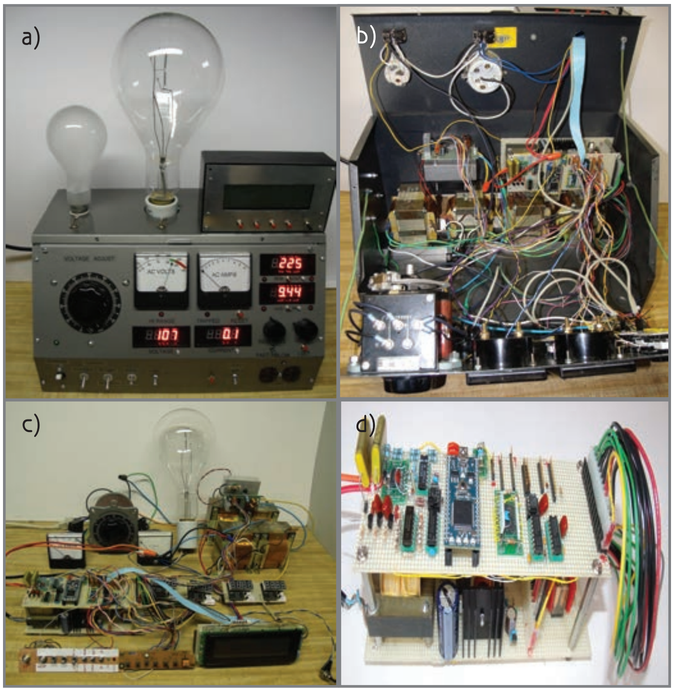



Photo 1a—The AC Tester powered up and running. The E meter is in the plastic case on the top of the Tester. The series current limit bulbs are on the top. b—Take a look at the inside of the AC Tester. The power transformers are in the back on the bottom. The triac assembly is above on the left. The power supply and processor board are above on the right. c—This is the prototype before I put it in a case. d—The bottom board is the power supply. The top board includes the LPC1768 processor and analog electronics.

are used to display current, voltage, trip current, and response time. A Texas Instruments PCA9539 I²C – I²C I/O expander is used to provide switch inputs, port outputs, and key press interrupts. Another PCA9539 provides a dot-matrix LCD interface for the E meter. An NXP Semiconductors PCA9507 two-wire serial bus extender bridges the 3.3-V and 5-V I²C buses. Five analog-to-digital converter (ADC) channels are used to measure AC current, line voltage, load voltage, trip current setting, and response time setting. The digital-to-analog converter (DAC) is used to set the latest high-current peak value for the overcurrent comparator. The SPI bus is used to communicate with an Analog Devices ADE7753 energy metering IC.

The mbed board enabled rapid development. It provided a pin-in-hole adapter for the LPC1768 for easy hardware prototyping. It also provided a function for nearly every I/O requirement. This cut down the time to get the LPC1768 up and running, as well as the time required to write the final program.

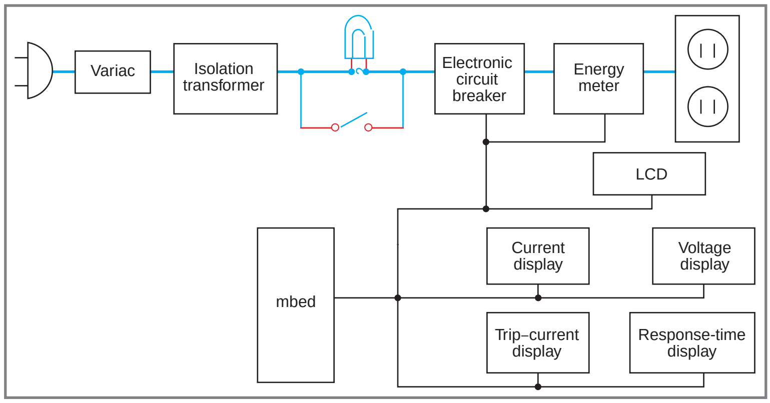



Figure 1—Input power is run through a variac and an isolation transformer. It then flows through current-limiting lightbulbs, an electronic circuit breaker, and an optional energy meter. The mbed processor measures the voltages and currents for the digital displays. It also controls the electronic circuit breaker.

## AC POWER INPUT

The basic function of the AC Tester is to provide a variable, isolated, and protected source of line voltage for testing, debugging, or repairing equipment. Figure 2 shows the power input section. This would be the most basic form of the AC Tester without any of the added features provided by the microcontroller. The line voltage comes in through a switch circuit breaker. It feeds the variac for voltage adjustment. The LED power indicator D1 uses the capacitive reactance of C1 instead of a resistor that would produce heat and require a high wattage. The variac bypass switch enables the variac to be bypassed for supplying nominal line voltage. This also enables the variac to be lower current, if desired. In that way, you can use it for low-current testing and bypass it for high-current work. This would enable you to use a less expensive lower-current variac. A Normal/Boost switch enables the line voltage to be raised above the nominal level. The line voltage then flows through an isolation transformer.

I used a pair of UPS power transformers to keep costs down. UPSes with bad batteries are inexpensive and commonly found. I used the transformers from four identical 600-VA units. They are wired back to back to provide 1:1 isolation. They can be switched to parallel for higher current or series for higher voltage (220-V) operation. Microwave oven transformers are also a possibility for the isolation transformers. You would have to remove the high-voltage secondary and magnetic shunt, and then wind a suitable VA secondary. The tradeoff would be the number of turns for a higher voltage that would enable a smaller wire for lower current. The line voltage then passes through M1, an analog AC ammeter.

Analog and digital meters both have their places. The analog meters provide for quick observation and indication of fast dips or oscillations. They can provide a significant amount of information

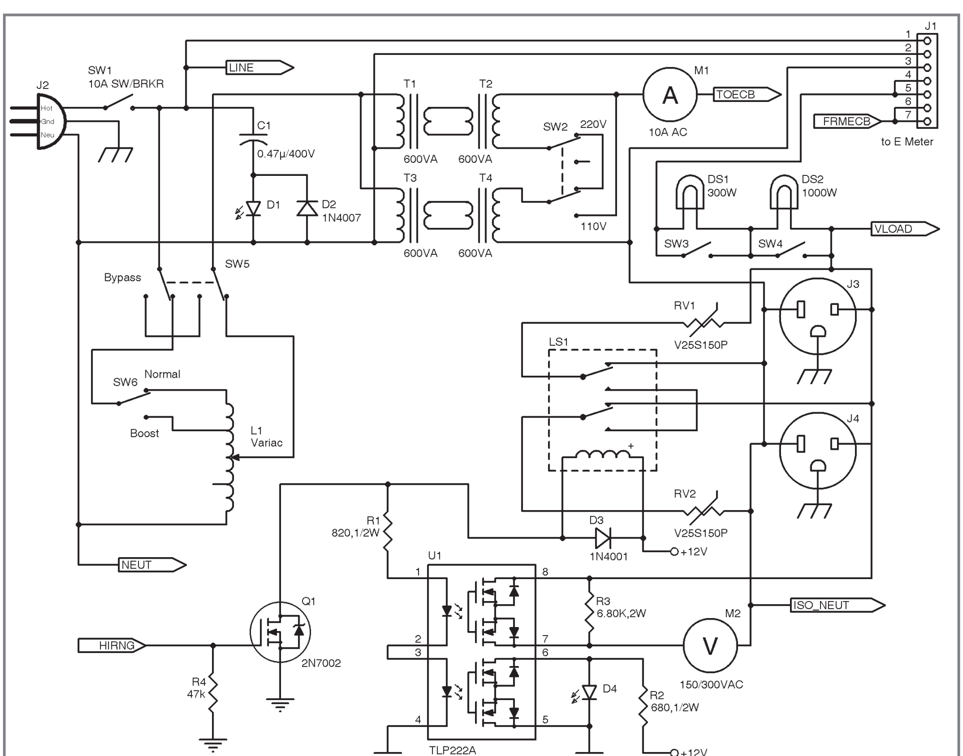



Figure 2—The AC Tester power input section includes the variac, isolation transformer, current-limiting lightbulbs, and meters. This is the most basic form of the AC Tester without any frills.

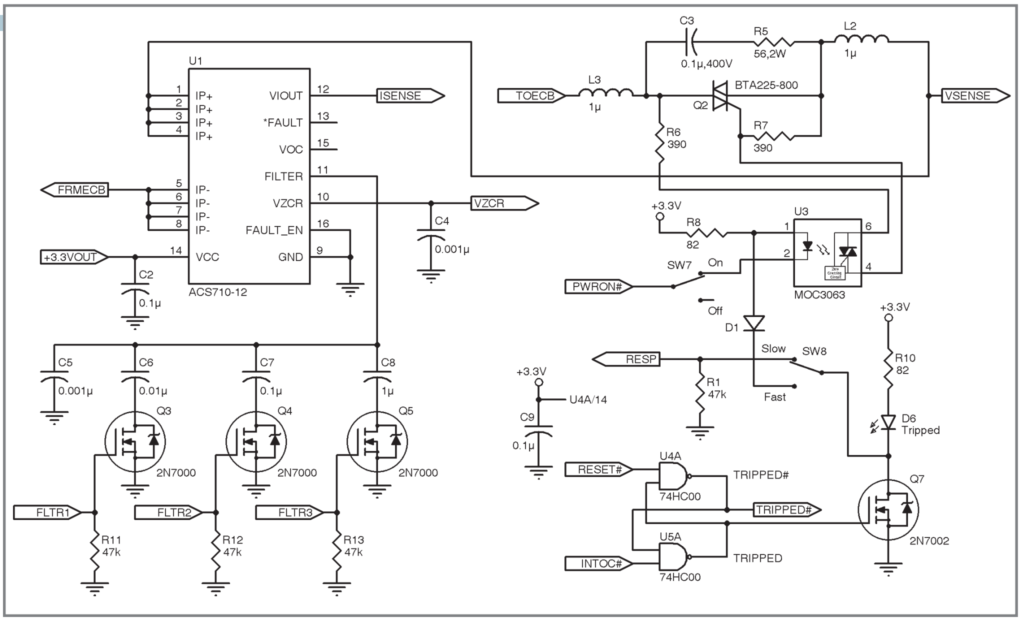



**Figure 3**—The current sensor and electronic circuit breaker section. A triac Q2 is used for the circuit breaker switch. U2 is the ACS710 Hall-effect analog current sensor with noise filter switches Q3–5.

once you've experienced them. You can't beat digital meters for accuracy. The AC Tester has both for that reason. You can run the line voltage through a 300-W or 1,000-W incandescent lightbulb to provide a current-limiting function. This provides some level of protection for a device being tested. It also again provides a quick visual indication of the current by observing the brightness of the filament. Optionally, the line voltage can go through the Hall-effect current sensor and the E meter. A duplex outlet provides the connection to the device. A red/green bicolor LED next to the outlet indicates if power is on or off. A MOV surge arrester RV1 and RV2 is also provided to absorb any spikes from the equipment being tested. To enable a 110-V or 220-V operation, a range switch switches the MOVs and voltmeter ranges. A high-range LED next to the voltmeter serves as a visual reminder of which scale to read.

The voltmeter range change for my meters is simply shorting out a series resistor R3. I used iron vane meters, which are not very efficient (about 1 W) or accurate but work on AC or DC. They were so inefficient I needed a high-power series resistor to double the range of the AC voltmeter. If you use a D'Arsonval meter with a bridge rectifier, you can use a simple voltage divider. Meters are fairly expensive. Although I found these at a Hamfest, they weren't what I needed. D'Arsonval meters are easy to spot because they will be heavy and magnetic. Iron vane will be light and nonmagnetic. The voltmeter was 50 VAC and the current meter was 5 A AC. I was able to add series resistance to increase the AC voltmeter range and rewound the current meter. With an iron vane current meter, this was about seven turns of 16-gauge wire. The next step is to change the faceplates. Tonne Software provides a nice Windows-based meterscale drawing program you can use to easily make new faceplates. It enables nonlinear scales (which iron vane are). You can even add color to the scales. I could then make an accurate faceplate for the rewound ammeter which enabled the nonlinear AC display. I was also able to add red and green scales to the voltmeter.

## CURRENT SENSOR & CIRCUIT BREAKER

The current sensor and electronic circuit breaker are shown in Figure 3. A Hall-effect current sensor U2 is used to measure the current. The Allegro MicroSystems ACS710-12 current sensor IC has some special features. It has internal circuitry to provide a fault circuit. The problem is that you can set it only for a narrow range. You can set the 12.5-A part to trip only from 14.7 A to 23.5 A. It can read currents up to 37.5 A, but it can only sustain currents of 12.5 A continuously because of the small package pins. This is why the circuit breaker function is done externally. It has an internal noise filter. The LPC switches in various filter capacitors C5 through C8 based on the response time for which the circuit breaker is set.

A snubberless triac Q2 is used for the switch. A snubber including R5 and C3 was added anyway for extra protection. Hash filter inductors L2 and L3 are also used even with zero-crossing switching. A zero-crossing opto triac gate driver U3 is used to provide isolation and control. A Fast/Slow response switch selects direct hardware control for fast response or adjustable slower response by software and the LPC.

## CURRENT CONDITIONING

The output of the ACS710 is run through an op-amp rectifier U6A and U6B to eliminate diode voltage drop (see Figure 4). The ACS710 provides a 56-mV/A output centered on the midpoint supply voltage. It is also a ratiometric current sensor. The 3.3 V from the mbed board is used to supply the 3.3 V to the ratiometric current sensor. This way the LPC ADC and ACS710 will track. The ACS710 provides a midpoint voltage reference VZCR. This reference is subtracted out and the sense voltage is amplified by two by U8A. This provides a positive-only full-range signal to the ADC and comparators. The overcurrent comparator U7C compares the OC SET value with the sensed current and interrupts the LPC when an overcurrent occurs. The peak current comparator U7D compares the sensed current with

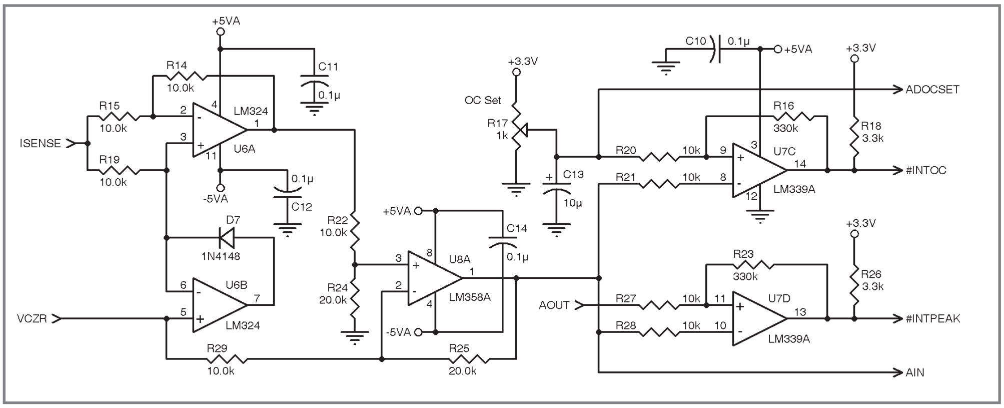



Figure 4—The current-conditioning circuit features rectifier U6A and B, midpoint voltage subtractor and amplifier U8A, overcurrent comparator U7C, and peak comparator U7D.

the DAC output from the LPC. Firmware raises the DAC (AOUT) voltage to correspond to the highest peak. Both comparators use hysteresis provided by the 330-kΩ resistors from the outputs.

## SAMPLE & HOLD

A peak and average current is provided to the ADC (see Figure 5). The average current path is through another rectifier U9A and rectifier D8. R31 and C16 provide a filter for the rectified AC. The peak current path is through a sample-and-hold U11. A sample and hold is used to ensure that the current peak is not lost by the time an A/D conversion can take place. An analog multiplexor is used to select the source for the current measurement because I ran out of ADC channels on the mbed. A one-shot U41 is used to provide the sample pulse. It is triggered by the peak comparator U7D.

## INPUT ATTENUATOR

The AC input is rectified by D12, filtered by C23, and attenuated by R33 through R37. It is buffered by a unity gain op-amp U6D. The line voltage is isolated, but a fuse F1 is still used for the isolated neutral-to-ground connection. I decided that it was easier to turn the input voltage into DC than to try to capture the peak of the AC waveform and sample it with the ADC. Using the zero-crossing interrupt and timing over to where the peak of the waveform should be with software was not giving me accurate results. Several resistors are used in series for the attenuator and the exact values needed are not standard. The tolerance error adds with the multiple resistors. To eliminate the need for high-accuracy resistors, I use a calibration value constant in the software instead.

## mbed PROCESSOR

The mbed processor U13 is nearly fully utilized. I didn't use the USB, serial, Ethernet, and battery backup. They could be useful for data acquisition, but it was not time for that to be developed.

The 40-pin DIP packaging was handy for prototyping and is actually space efficient as I found out. Just before the NXP mbed Design Challenge contest deadline, Murphy's Law occurred once again. A high-voltage spike worked its way back into the mbed and totally toasted it. I had planned on making a dedicated processor board and saving the mbed board for other projects requiring a quick turnaround. This unfortunate event forced me to do so. I soon found out that trying to put the LPC1768 on an adapter with its support hardware was much bigger than the mbed board. My processor board has the LPC1768 with its decoupling capacitors and a 3.3-V regulator. It has a 12-MHz crystal for the processor clock; a seven-pin programming connector (to bring out RESET, TXD, RXD, and NMI), and power and ground. I included the four mbed LEDs. As you can see in Photo 2, the end result is bigger than the mbed board.

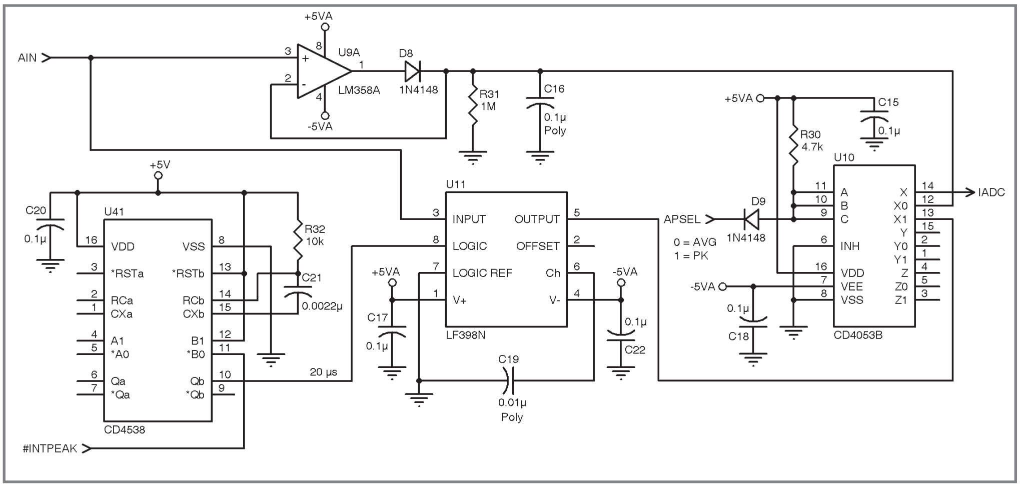



Figure 5—The condition current signal has two paths. The average path runs through rectifier U9A and is filtered to DC by C16. The peak path goes through a sample-and-hold U11. The mbed processor selects the two signals through analog multiplexor U10. U41 provides a sample pulse to the sample and hold. It is triggered by the peak comparator U7D.

I replaced the LPC1768 on the bad mbed board, but the other custom USB processor must also have been bad. With the new LPC, the power LED was back on, but I could not communicate with the PC over USB. I decided to take advantage of what was left of the board and use it as my processor board. I added the seven-pin programming connector and cut the traces to the bad USB processor.

My first attempt at a programming board was a simple RS-232 interface. It has a Reset push button and a Program jumper. The level translator is a Maxim Integrated Products MAX3232 transceiver powered from the programming connector.

My next programming board was a USB board. It was a simple board with a Future Technology Devices International (FTDI) MM232R USB serial UART development module providing the USB-to-RS-232 translation. It's powered from the USB interface. It's small, but the MM232R is a little pricey.

My final programming board used an FTDI FT232 UART interface. This board needed more components but the device was inexpensive, and I liked the ability to have the RX and TX LEDs to give an indication of what is going on. It is also powered from the USB interface.

Check mbed.org for information on replacing the mbed module with an LPC1768. I compiled the program with the mbed compiler and saved the .bin file to a folder. Remember to use a simple name that's less than eight characters. The reason for this is the bin2hex program is run under DOS, which gets confused with longer file names. Download, extract, and run hex2bin.com's bin2hex program. Then run Embedded Systems Academy's Flash Magic tool. Set up your COM port and input the crystal frequency. Make sure the ISP jumper is on the programming board and cycle power to the LPC or press the Reset button on the programming board. Do a blank check to see if you can communicate with the LPC. Browse to the saved hex file location and select it. Click the Start button and the LPC will be programmed with the compiled binary code. Remove the ISP programming jumper on the programming board and press Reset or just remove the ISP board and cycle power. The LPC should now be programmed with the binary file and running.

## LEDS, SWITCHES & BUTTONS

SAA1064 I²C LED drivers U16, U19, U22, and U25 drive the LED displays. The four driver chips are assigned individual I²C addresses by multilevel

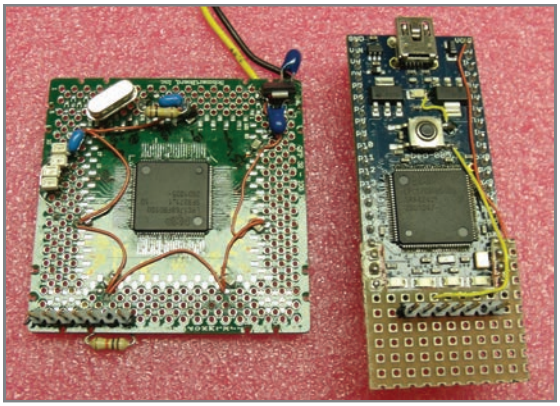


Photo 2—Here's a close-up view of the main electronics board. The ACS710 is on the left side of the board. Next is the analog signal-conditioning portion. The mbed LPC1768 is located in the middle. The PCA9507 I²C-level converter is just above the mbed. The PCA9539 I²C-to-parallel port for the switches is next. The sample-and-hold hardware is on the right.

strapping of the address lines. They are inexpensive and do a great job of brightly driving the LEDs with a minimum of external components.

The various switches and push buttons would require more I/O than is available on the mbed, so they are also $I^2C$ attached by a PCA9539 U26. The PCA9539 provides an interrupt to the processor if any of the inputs change state to free the software from polling.

## $I^2C$ ZERO CROSSING & POWER SUPPLY

The PCA9507 U27 provides $I^2C$ bus-level translation between the 3.3-V processor $I^2C$ bus and the 5-V level external $I^2C$ bus. The external bus is run at 5 V for $I^2C$ adapter chip reasons as well as to provide a higher noise immunity for the off-board connections.

Voltage and current zero-crossing detectors U7A and U7B provide interrupts for zero crossing to the processor. The comparators have hysteresis provided by the feedback resistors R61 and R66. This prevents noise pulses near the zero-crossing points. A negative offset is summed into the signals to ensure a good zero-crossing pulse.

The power supply supplies 12-V, 5-V digital, 5-V analog, 3.3-V, –5-V analog, and the AC voltage zero-crossing reference (see Photo 3). The 5-V digital supply is a switching power supply U29 to supply the higher current required by the LED displays. A PFET Q16 is used as the switch to provide less voltage drop than a PNP transistor $V_{CESAT}$. The transient suppressor D23 is there to protect the output in the event of a PFET short. A separate rectifier and filter supply the 5-V analog linear regulator U30. Linear regulators also supply the lower currents required for the 3.3-V power supply U31 and the analog 5-V U30.

## FAN CONTROLLER & BEEPER

The fan controller U33 is an LM75 $I^2C$ temperature sensor that is thermally attached to the triac heatsink (see Photo 4). It is soldered on a small SMT fanout adapter board. The board is mounted with the LM75 against the heatsink with

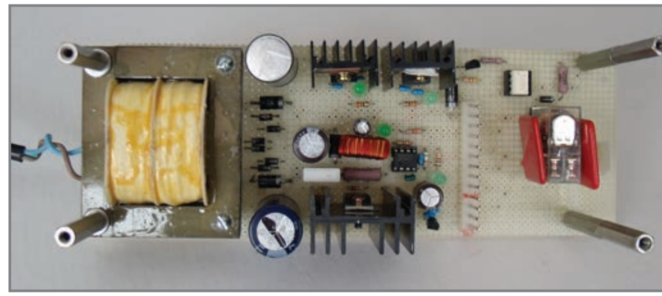


Photo 3—The power supply board. The 5-V switching regulator is located on the bottom in the middle. Above it are the 3.3-V and 5-V linear regulators. The 110-V/220-V MOV relay is on the right.

thermal tape. The over temp open drain output drives a PNP transistor Q17 to turn on the fan. This cuts down on the audible noise. The triac Q2 only gets hot enough to need the fan under heavy loads.

## E METER

The ADE7753 E meter is a 5-V device (see Photo 5). The 74LVC4245s U34 and U35 provide 3.3-to-5-V level translation between the LPC1768 and the ADE7753. The E meter is on a separate module to make it optional. Pin 13 of connector J12 is grounded in the E meter and indicates back to the microcontroller that the E meter is attached.

A Hantronix HDM64GS24 LCD is $I^2C$ attached to the processor by a PCA9539 U36. This provides the 14 I/Os required since there were no more left on the mbed. Doing a parallel bus LCD with a serial $I^2C$ interface is slow and it does have painfully slow response time for full screen clears or writes. To try to minimize the slow response time, firmware only does a screen erase on power-up. It then writes all the screen text. After that, only values are updated. Since JFETs are normally on with no gate drive, they work well to switch off the LCD bias until 5 V comes up and also during LCD controller resets. This is recommended to prevent damage to the LCD. Q21 senses the loss of 5 V and grounds the emitter

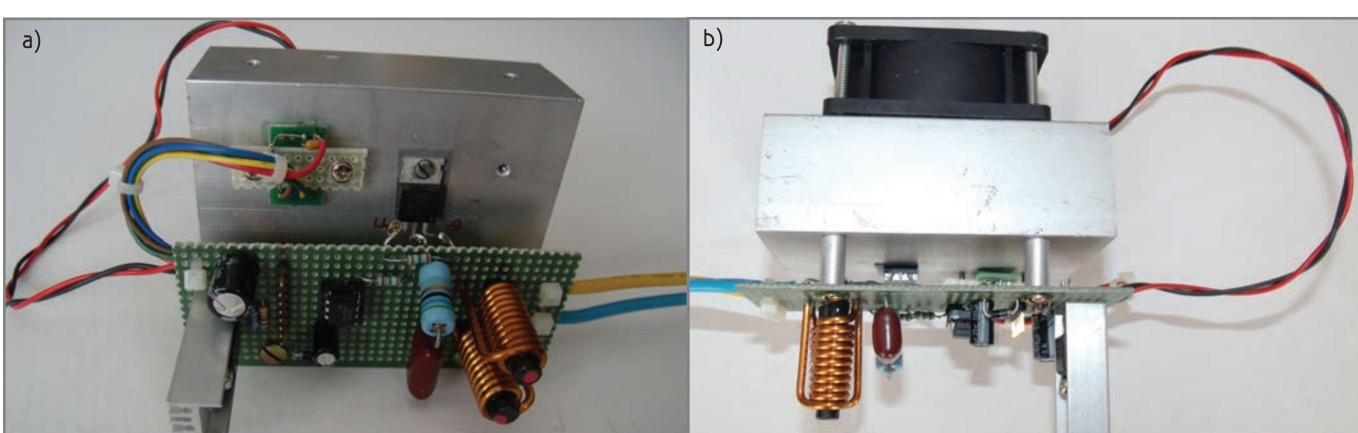



Photo 4a—The fan board assembly. The LM75 temperature sensor in the upper left-hand corner is mounted on an SMT adapter board and attached backwards with thermal transfer tape between the top of the IC and the heatsink. The triac is mounted with a SIL PAD insulator to the heatsink. The triac driver circuit and hash filter are on the right of the perf board. The fan driver transistor is the heatsink on the left of the board. b—The fan is mounted against the heatsink fins. It is attached to the heatsink by round spacers between the heatsink fins. The electronics board also bolts into them from the back with additional spacers.

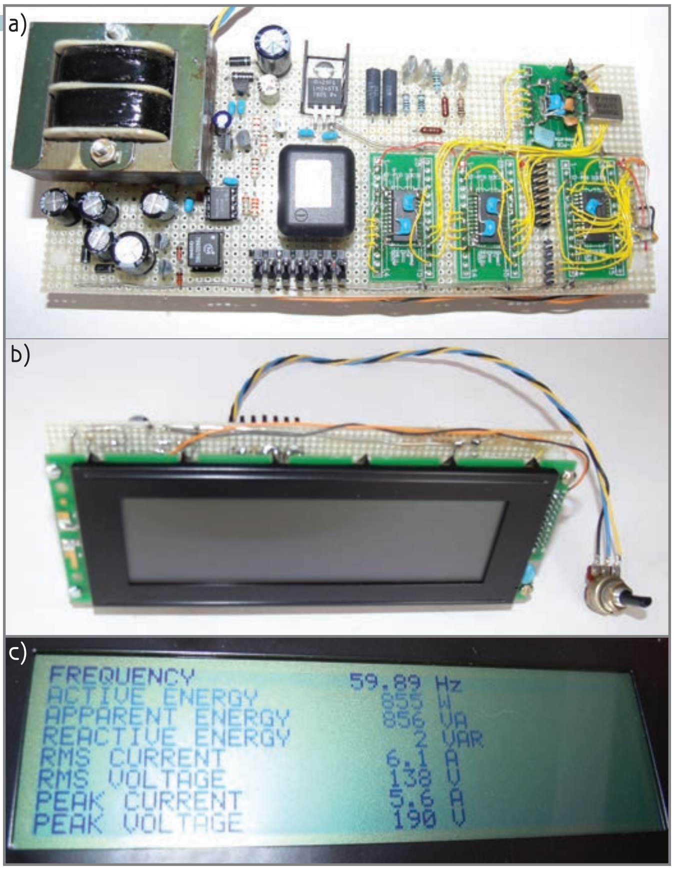



Photo 5a—The E meter board. The power supply section is on the left. The large black cube is the CS2106 current transformer. The three SMT ICs mounted on adapters are the PCA9539 I²C-to-parallel adapters. The ADE7753 is in the upper right corner. b—The E meter assembly consists of the HDM64GS24 LCD with the E meter electronics board bolted to the back. c—This actual E meter display shows frequency, current, voltage, and energy.

follower Q19, removing the negative bias from the LCD. Q21 also does this during an LCD reset. Emitter follower Q19 is for contrast adjustment. R79 provides current limiting of the bias voltage when Q20 or Q21 clamps to ground.

The E meter uses an ADE7753 energy meter IC (U37) intended for utility watt-hour meters. It provides an easy way to measure watts, VA, VARs, and so forth. It also provides peak current and voltage and RMS current and voltage measurements. In addition, it has a high-resolution frequency output. The input voltage attenuator (R86, R87, and R88) is 100:1 to provide 500 to 0.5 V to match the maximum input range of the CH2 ADC. The burden resistor R82 and R84 on a Coilcraft CS2106 current transformer T5 has been reduced to 22.6 Ω to match 15 A to the 0.5 V input of the CH1 ADC.

The backlight power supply generates the 120 V, 400 Hz for the electroluminescent backlight. A voltage quadrupler D33, D34, D35, and D36 multiplies 36 VAC to around 160 VDC. A 555 timer U38 provides a 400-Hz square wave to an inexpensive Murata Power Solutions 78601/16C pulse transformer T7. The pulse transformer alternately triggers the series-connected SCRs, Q22, and Q23 to charge and discharge the capacitive backlight. It generates about a 120-V square wave.

The E meter uses a simple power supply. The current requirements are very low, so half-wave rectifiers D39 and D40 are used with light filtering C87 and C91 (see Figure 6). This was to accommodate the higher voltage transformer T8 in order to get the 36 VAC for the backlight multiplier. Linear regulators U39 and U40 supply the E meter 5 V and the LCD bias –15 V.

## FIRMWARE

The firmware scans the controls and updates the displays. It scans the E meter registers through the SPI bus. Frequency, watts, VA, VAR, RMS current, RMS voltage, peak current, and peak voltage are all displayed on the I²C-connected LCD. In Slow Response mode, the circuit breaker function is under control of the processor. If the overcurrent trip point is reached, a timer is started. The timer is loaded with the response time set by a potentiometer that is read by an ADC channel. The OC condition is monitored and if it is present until the end of the response time, the circuit is opened. A Reset button is provided to reset the circuit breaker. A peak current display is updated by a peak current interrupt. The peak current is read from the sample and hold by the ADC. This value is written to the DAC as the new input maximum for the hardware comparator. The AC voltage and AC load voltage are also read by the ADC and displayed.

## OPERATION

Operating the system is fairly straightforward. The main

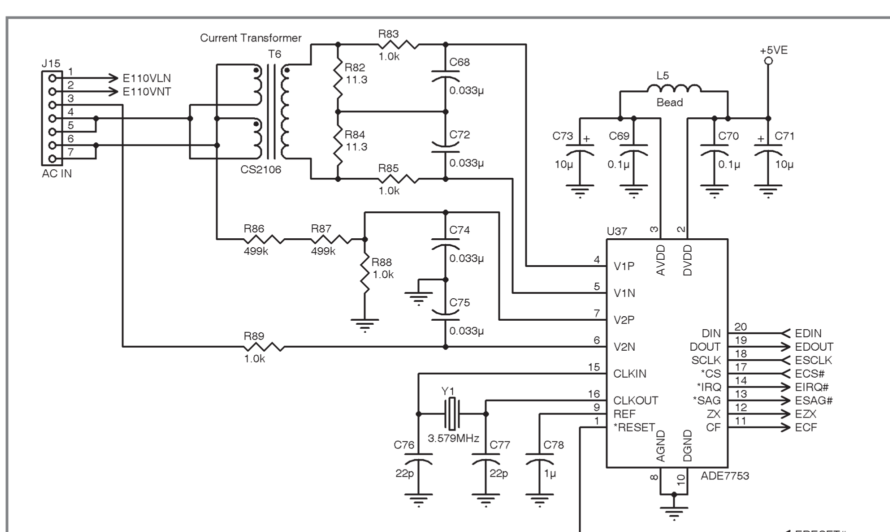



Figure 6—This is the energy meter U37. It talks SPI back to the processor. Current transformer T6 samples the load current. R86, R87, and R88 are the voltage-sample attenuators.

circuit breaker turns on power to the tester and protects it from overcurrent. A variac bypass switch enables the unit to be operated with nominal line voltage or through the variac. A boost switch enables the variac to increase the line voltage above the nominal. The variac enables adjustment of the line voltage. A 110-V/220-V switch enables selection of 110 V or 220 V. A 300-W or 1,000-W series-connected lightbulb can be switched in for another form of current protection and indication. For very low power devices, a 100-W bulb could be used in place of the 300 W.

Analog AC voltmeter and ammeters show quick indications of voltage and current. Digital LED voltage and current meters provide accurate displays. The overcurrent potentiometer selects the over-current trip point. The minimum value is about 1 A due to sensor noise. The response time potentiometer sets the circuit breaker delay in Slow mode. A Fast/Slow mode switch selects a fast hardware circuit breaker function or a slow microcontroller circuit breaker. The On/Off switch removes power from the triac drive to turn the output on or off. The response time/$V_{OUT}$ switch selects the response time or load voltage to be displayed. The load voltage display is useful when the series lightbulbs are used. It will give you an indication of the drop across the bulb, which may become excessive. The Average/Peak switch selects tripping on a sustained average current or a peak current. This switch is only used in Slow Response mode. The Reset button resets the breaker after a trip. The I Set button displays the overcurrent trip point setting while depressed on the peak current display. The peak current display shows an accumulated highest current. The Peak Reset button resets this display. This is useful for seeing turn on surges or for devices that have intermittent problems. The E meter has an LCD to display watts, VA, VAR, peak voltage, peak current, RMS voltage, RMS current, and frequency.

## CONSTRUCTION

I built the tester in a steel case because of the weight of all of the power components. The four power transformers are bolted to aluminum angle as a stiffener for added support. Most of the power parts are attached to the steel front panel.

The electronic assemblies are put together on perf board with surface-mount adapters for the surface-mount parts. I found it handy to size the grounding wire from the case to the top-hinged panel to the appropriate length to hold the lid when opened. I did the same thing with the two grounding wires from the main chassis to the front panel. This holds the front panel when working on the inside.

The E meter section would not fit in the tester box. I placed it in a plastic project box and bolted it on the top of the tester. Since the E meter is an option anyway, it seemed like a good solution.

## FUTURE IMPROVEMENTS

The E meter has many other functions—such as actual watt-hour measurements—that I'd like to incorporate. I'd also like to include a power factor display. The intention is to use the Function push buttons with the LCD to add additional screens with additional information. The LCD's graphics capability would also be useful to graph current spikes and so forth. The USB and Ethernet with the mbed-provided functions would make a very nice data acquisition tool.

*Kevin Gorga (kgorga@stny.rr.com) has an MSEE and has been a design engineer with IBM in Endicott, NY, for the past 35 years. His technical interests include embedded system design, power systems, and motor controls.*

### PROJECT FILES

To download the code and a full set of schematics, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/263.

### RESOURCES

Embedded Systems Academy, Inc., Flash Magic, www.flashmagictool.com.

hex2bin.com, www.hex2bin.com/bin2hex.

mbed, "mbed NXP LPC1768: Getting Started," http://mbed.org/nxp/lpc1768.

———, "Cookbook: Prototype to Hardware," http://mbed.org/cookbook/Prototype-To-Hardware.

Tonne Software, "Meter and MeterBasic: The Windows Programs for Drawing Analog Meter Scales," 2011, www.tonnesoftware.com/meter2.html.

### SOURCES

**ACS710-12 Current sensor IC**
Allegro MicroSystems, Inc. | www.allegromicro.com

**ADE7753 Energy metering IC**
Analog Devices, Inc. | www.analog.com

**CS2106 Current transformer**
Coilcraft, Inc. | www.coilcraft.com

**MM232R USB Serial UART development module and FT232 UART interface**
Future Technology Devices International Ltd. | www.ftdichip.com

**HDM64GS24 LCD**
Hantronix, Inc. | www.hantronix.com

**MAX3232 Transceiver**
Maxim Integrated Products | www.maxim-ic.com

**mbed Microcontroller board**
mbed | www.mbed.org

**78601/16C Pulse transformer**
Murata Power Solutions, Inc. | www.murata-ps.com

**LPC1768 Microcontroller, PCA9507 two-wire serial bus extender, and SAA1064 LED drivers**
NXP Semiconductors | www.nxp.com

**PCA9539 I²C – I²C I/O expander**
Texas Instruments, Inc. | www.ti.com

by Mohammed Dogar

# Renesas RL78

## The True Low-Power MCU Platform

In our green energy world, many electronic designs are driven by requirements for reduced size, improved scalability, intelligent functionality, and most importantly, low power consumption.

The Renesas RL78 microcontroller (MCU) family is specifically designed to meet these demands by incorporating the highest peripheral integration, intelligent CPU architecture, and advanced power-management capabilities to enable "true low-power design."

In addition to excellent general low-power characteristics, RL78 MCUs incorporate special functions to minimize operating current. Major sections of the MCUs can be turned off, while key peripheral blocks of the device continue to function.

This smart low-power operation is achieved with the unique Snooze mode capability. Snooze mode dramatically reduces the power consumption of many typical MCU functions. The power savings are accomplished by allowing common data acquisition or data-transmission functions to operate without the need to wake up the CPU. This operational flexibility is a significant advantage over other low-power modes in which the CPU must remain active and assist with common peripheral functions.

In a system that periodically measures an analog signal, the Snooze mode enables an RL78/G13 MCU to achieve more than 30% reduction in the system's average power dissipation compared to an implementation without this mode.

Besides Snooze mode, RL78 MCUs have other important low-power characteristics that are valuable for power-constrained designs. Their wide operating range, from 1.6 to 5.5 V, suits battery-based applications where the voltage ($V_{CC}$) drops over time as the battery gradually discharges.

### REDUCED POWER CONSUMPTION

In extreme applications, some battery-operated equipment must run off a battery for its entire operating lifetime, without any recharges whatsoever. Such designs require the lowest operating current possible. It is absolutely critical to turn off functions in the system whenever they aren't needed and wake up functions only when they're required. The ability to wait in a very low-power state until action is required and then wake up to take necessary action, and do so while using as little current as possible, can dramatically extend useful lifetimes.

Renesas's RL78 MCUs are recommended solutions for embedded systems that mandate low-power operation requirements because they provide advanced power-management capabilities.

These functions enable the MCU to run with exceptional power efficiency in the normal Run mode; disable CPU operation, saving power in Halt mode (enabling fast CPU wake-up time); disable more of the MCU functions in the Stop mode to save the most power (at the expense of a longer CPU wake-up time); and deliver even greater power savings with Snooze mode. Figure 1 depicts an operational flow diagram for these three modes.

### SNOOZE MODE OPERATION

The Snooze mode enables some peripheral functions to wake up and execute simple operations while the rest of the MCU is stopped. This saves a significant amount of power compared to the Run or Halt modes because in Snooze mode, the CPU is off and only the peripherals that must operate are enabled.

Data reception from the synchronous serial port, the UART, or a data conversion by the analog-to-digital converter (ADC) can operate in the Snooze mode by waking up the associated port, but not the CPU.

The ADC can "wake up" when the real-time clock (RTC) or the interval timer generates interrupt signals to start a conversion. Similarly, the synchronous serial port can "wake up" when the serial clock input pin edge is detected, and the UART can "wake up" when an edge on

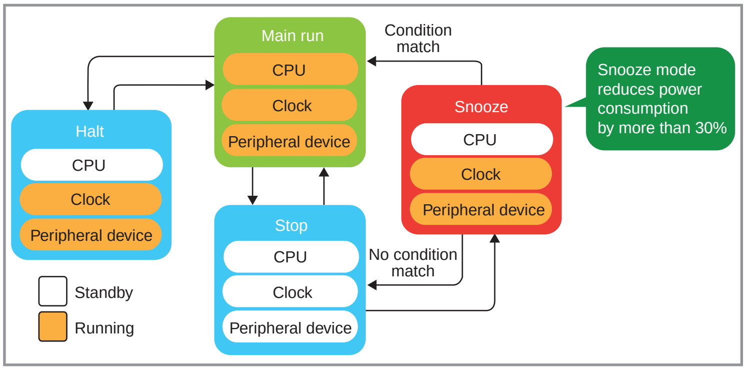



Figure 1—An operational flow diagram of the MCU functions

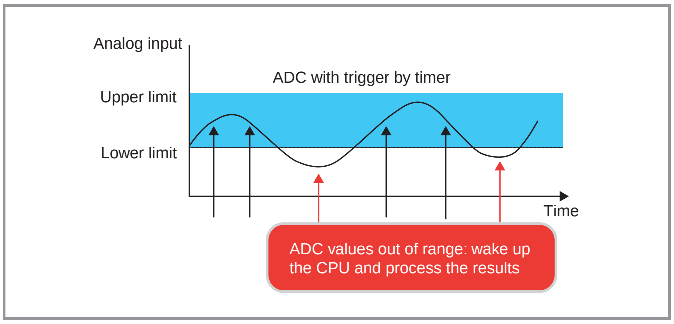



**Figure 2**—Snooze mode using the ADC triggered by a timer

the RXD input is detected.

After any data reception operation in Snooze mode is completed, a "match condition" is checked. If the condition is a match, then the MCU exits the Snooze mode and enters the Run mode. If the condition isn't a match, operation returns to the Stop mode. Thus, the CPU can be activated only when the data received requires action from the CPU. Figure 2 depicts Snooze mode using the ADC triggered by a timer.

For example, because A/D conversion uses only 0.5 mA in Snooze mode, rather than 5 mA in Run mode, dramatic power consumption advantages are obtained. Thus, an A/D conversion can be performed in Snooze mode using only 0.5 mA, 90% less than the 5 mA required to make a conversion in Run mode.

## NOW IT'S YOUR TURN

I discussed the ultra-low-power features of the RL78 MCUs, now what can you do with them? How will you use the energy-saving features of the RL78 in your project for the RL78 Green Energy Design Challenge? What's more, how can you incorporate these features into your everyday design and push the green energy envelope to the next level, working to shape the future?

*Mohammed "Mo" Dogar works for Renesas Europe as a product marketing manager, focused on promoting and selling microcontroller devices to major OEMs and distributors. Mo's group defines and develops MCU product roadmaps and development support infrastructure for wide range of industrial and consumer electronics applications.*

*To register and enter the RL78 Green Energy Challenge, go to: www.circuitcellar.com/renesasrl78challenge*

# QUESTIONS & ANSWERS

# Industrial Control Engineering

## An Interview with Aubrey Kagan

*Aubrey Kagan's engineering experiences and work have run the gamut from studying in Israel to designing controllers for mines in Africa to helping create specifications for the remote control arm on the International Space Station. He is now an engineering manager at a Toronto-based design house for industrial control interfaces and switch-mode power supplies. Aubrey and I recently discussed his background and some of the interesting projects he has designed and written about over the years.—Nan Price, Associate Editor*

**NAN: Where do you live? What city, country?**

**AUBREY:** I live on the northern edge of Toronto, Ontario, Canada. However, that belies my accent, which the readers obviously cannot hear. I was born and grew up in "deepest, darkest Africa" just north of Rudyard Kipling's "great gray-green, greasy Limpopo River" (see "How the Elephant Got Its Trunk" from Kipling's *Just So Stories*) in what is now called Zimbabwe (then Rhodesia). I did my undergraduate engineering degree at the Technion, Israel Institute of Technology, and then returned to Africa for my MBA at the University of the Witwatersrand in South Africa. My early years in engineering were spent in South Africa, immigrating to Canada in 1989.

**NAN: What is your current occupation?**

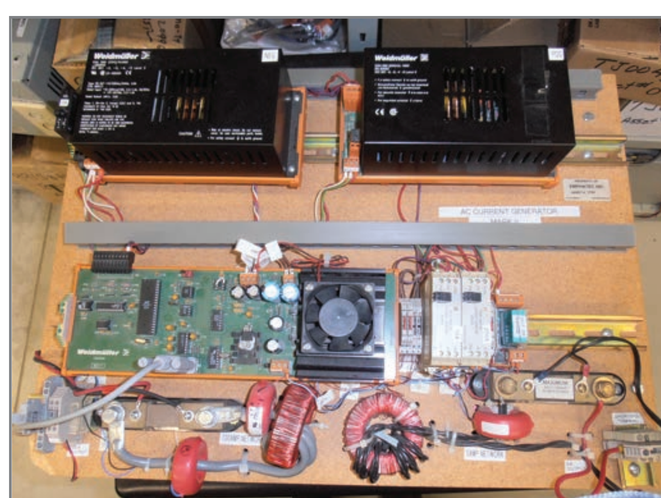
**AUBREY:** I am an engineering manager at Emphatec, although managing occupies only a small portion of my day—the majority of my time is engineering. Most of the projects are for industrial monitoring and control. They tend to be a blend of analog and digital approaches and usually are quite compact with only a single function.

**NAN: How long have you been interested in designing embedded systems?**

**AUBREY:** I was given the opportunity to get into embedded design long before anybody thought to call it that. It was in 1977, and all we had were microprocessors, which I was trying to design into HF radio transceivers. I had been struggling with phase lock loops and control of the frequency divider seemed a likely candidate for computer control. Just at that time, there was an article in *Popular Electronics* on creating an evaluation board for the RCA CDP1802 COSMAC microprocessor. I used that as the basis for the development and as they say, the rest is history.

**NAN: *Circuit Cellar Online* featured your article, "Developing an AC Current Generator" (119, 2000). Tell our newer readers about that project. Do you still use the generator? Have you made any upgrades to it?**

**AUBREY:** That was my first *Circuit Cellar* article and my only collaborative effort (with Ernesto Gradin). It is probably my favorite project because it is so unusual and remains pertinent to this day. Some of the products that we make involve monitoring an AC current and converting the measurement to a 4-to-20-mA analog signal. Some of the devices will measure currents up to 100 A AC. In order to test and calibrate these units, obviously you need an accurate current. If you use a variable AC voltage into a fixed load or a fixed voltage into a variable load to generate the current, you will be working with dangerous voltages and lots of heat. This leads to errors due to heating and more importantly health risks to the operators. We all




This AC current generator was one of Aubrey's favorite projects.

know in transformers ($V_{IN} \times I_{IN}$) = ($V_{OUT} \times I_{OUT}$) and $V_{IN}$ = (N × $V_{OUT}$) and so if you make a transformer with a low number of output turns, there is a low output voltage, and for a given power input you can then derive a high current—no heat and very low voltage. To improve the performance, we added a feedback loop with a micro then implemented PID control. The generator is still in use. I have not made any upgrades to it, but I certainly could improve upon it now. I would like to increase its resolution, and of course some of the components are now obsolete, so they would need revision. I might consider onboard displays as well, not control from a PC.

**NAN: Your 2002 article series, "Driving the NKK SmartSwitch" (*Circuit Cellar* 144 and 145), focused on using a Cypress Microsystems programmable system-on-chip (PSoC) microcomputer as an interface to drive the SmartSwitch. Tell us how this project came about.**

**AUBREY:** Signal conditioning modules in the process-control market tend to be physically small, typically 2″ high by 3″ deep by 0.75″ wide. Of course, there are many much bigger and smaller examples. All of them mount on a rail installed in a panel. Aside from some LED indications, there is very little information you can glean by just looking at the modules. As a result, there has been a slow trend in the industry to add displays to each individual module. Because of the size, the displays are small and are limited to seven-segment displays of up to four digits and sometimes some indicators, if a custom LCD has been used. Also, the displays are invisible when the panel door is closed. The NKK SmartSwitch would allow three lines of six alphanumeric characters and even some graphics. It would also allow the user to change operational parameters for the module. The NKK projects through the panel door and so the information is available to the outside world.

Simply driving the display was the focus of my discussion in *Circuit Cellar*. At the time, the article had the distinction of being used as an application note by two different companies simultaneously (NKK and Cypress).

But there is much more to the story. If an NKK SmartSwitch and driver were added to a single module, it would probably double the effective price of the module, and so we came up with a networked approach that allowed a single NKK SmartSwitch to be shared among up to 30 different modules spreading the costs and now becoming economically more viable.

**NAN: Five years ago in your article series, "Resilience in Embedded Designs" (*Circuit Cellar* 206–208, 2007), you covered many elements of design, including power supply, voltage supervisors, watchdogs, output, and software. Is there anything you'd add to that now?**

**AUBREY:** This was probably the hardest article I ever wrote. I have always tried to write about original topics, but this time I thought that since I had been around a long time, I had amassed a lot of knowledge that I had concentrated from many sources. I thought that if I put it in one place, it might be beneficial for those starting in the industry. I found that compressing the article into the series of three was a difficult and frustrating task. Would I add to it? There is a probably a book in what I would add, but I am not sure anyone would ever read it.

An NKK SmartSwitch connected to a power supply

**NAN: Your article "The 4-to-20-mA Current Loop," (*Circuit Cellar* 241, 2010) described a vast array of fieldbus options available for the 4-to-20-mA current loop. If you could revise this article now, would you add any more options?**

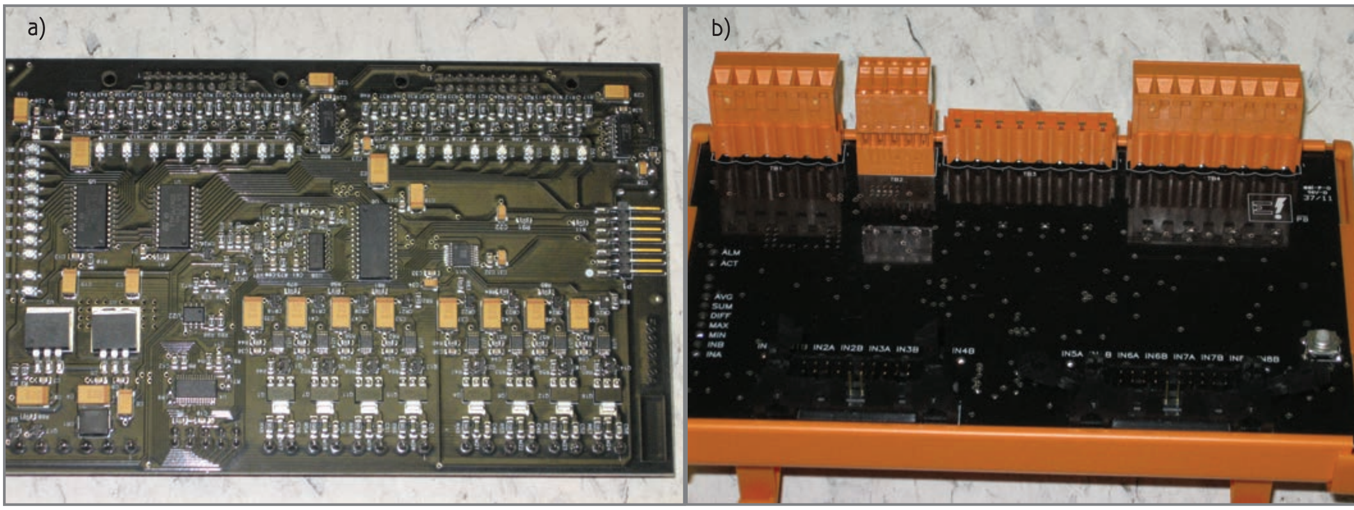
**AUBREY:** The current loop is old technology and so there is not much to revise. It just keeps rolling along. HART technology allows a frequency-shift keying (FSK) signal to ride on top of the 4-to-20-mA loop. This is an aspect that could be developed if I had a suitable project to drive it.

The digital fieldbus industry is in disarray and is losing ground to different Ethernet variations and any description from me would require that I acquire significantly more knowledge. It seems to me that EtherCAT may be a potential topic, but again I would need some motivation to move in that direction.

**NAN: Tell us about your interest in Excel. You wrote an article series, "Do You Excel in Electronics?" (*Circuit Cellar Online*, 138–140, 2002). Did this article series help inspire you to write your book, "*Excel by Example: A Microsoft Excel Cookbook for Electronics Engineers*"?**

**AUBREY:** Inspiration? Not exactly. It seems to me that book publishers scan magazines like *Circuit Cellar* looking for book ideas. In my case, I was approached by my publisher to see if I could turn the series of articles into a book. The article series certainly provided the basis, but I had never considered writing a book at that point.

My interest in spreadsheets began when I was running my own company back in South Africa. I was trying to cost products using Supercalc on an Osborne 1 computer. Those were the days… Not! Anyway, I discovered the tabular lookup feature and then realized that there were many other features




Aubrey's latest project is a 16-to-8-channel, 4-to-20-mA redundant loop controller. Here you see the bottom (a) and top (b).

that I could use in design. I developed several applications, and moved first to Quattro Pro and then Excel around the introduction of Windows 95. Just after that, I suddenly discovered that a spreadsheet could be used to solve equations. In my article on the 4-to-20-mA current loop, I actually describe the approach that was the seminal point of the article and book.

**NAN: Give us a little background about the book. What can readers expect to learn from it?**

**AUBREY:** Well, the book is no longer in "print." It is available in PDF electronic format from the publishers and on Kindle from Amazon. My initial intention was that since everyone had Excel for free, as it were, it could and should be used as a design tool. I am sure that the readers would take that away. But right when I started writing the book, I came across a design idea from *Circuit Cellar* contributor Alberto Ricci Bitti which fundamentally changed my approach. Excel is such a versatile tool that it allows you to create different approaches to solutions that you may never consider. I tried to convey this approach through all the examples in the book and I hope that the enthusiasm that I felt permeates through to the reader and provides inspiration to create interesting applications using Excel. (By the way, the Kindle edition does not include the Excel worksheets that were included with the book. If any reader does buy it, contact me and I will send them the workbooks.)

**NAN: In one of your latest blogs, "Overcome MCU Probing Challenges on Surface-Mount Boards," (*Microcontroller Central*, March 2012,) you discuss some of the challenges and solutions to working with small components. And in "Custom Sensors Enhance MCU Design" (*Microcontroller Central*, March 2012) you describe two instances where using custom-made sensors helped to solve some design issues. Can you tell us about some other challenges and solutions you have encountered?**

**AUBREY:** The biggest challenge is keeping up with the industry. When I started out, there was the huge flurry of development in the microcomputer world, but it was limited. Keeping up was manageable. Over the years, the market has exploded—8-, 16-, 32- and 64-bit micros with the multitude of interfaces. Keeping up with that is really tough for several reasons. Firstly, the absolute volume of information is enormous, but the delivery system of the information is changing. By now you will have gathered that I am old school and I prefer my data on paper. The magazines have gone digital and I just don't read them as closely as do on paper. Please, *Circuit Cellar*, keep the paper format! Secondly, the interfaces are really complex and each one is different. There used to be a UART and that was all. The communications protocol was up to you, Modbus was as complicated as it got. Now there is USB, Ethernet, TCP/IP, or Bluetooth, just to get started.

When I work on a project, there are several software applications that I use. There is the microcontroller compiler, schematic capture and layout, mechanical drawing, FPGA/CPLD software, Word, Excel, and maybe even more. Each one has a different user interface for even activities that you think would be common like text entry, selection of objects, or drawing lines. To say nothing of the interface between competing compiler/debuggers like changing between Cypress and Microchip Technology or when having to change CAD packages like PCAD to Altium. It is a great challenge being able to switch between applications and work effectively.

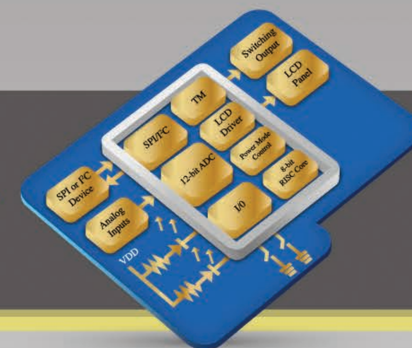
**NAN: Last question. Are you currently working on or planning any embedded-design related projects**

**AUBREY:** My most recent project involved a controller that adjudicated 16 channels of paired redundant 4-to-20-mA current loops into eight single 4-to-20-mA outputs for a programmable logic controller. My next project involves data acquisition from an industrial process and transmission to a portable computer (probably a tablet) via some kind of wireless transmission. ◾

HOLTEK
www.holtek.com

# All In One

**Ultra Low Power Consumption - Highly Functionally Integrated Flash MCU Solutions**




The Holtek Flash MCU range of devices encompass an extensive range of peripheral functions making them suitable for use in a diverse application area such as health care products, instrumentation products, household appliances, industrial control, consumer products, automotive peripheral products to name but a few. With one range types also including Holtek's unique TinyPower™ technology, the ability is provided to meet the demands of today's environmentally conscious products.

- **HT68Fxx** Internal Multi-function Timer Modules. Internal oscillators with four frequency selections. Multi-function I/O pins with re-mapping function. 4 LCD SCOM outputs for direct driving of LCD panels. Multiple Communication Interfaces.
- **HT66Fxx** Includes all the HT68Fxx functions with an additional 12-bit ADC and internal reference voltage source.
- **HT67Fxx** Includes all the HT66Fxx functions but also uses Holtek's TinyPower ultra low power technology and also includes an internal R-type/C-type LCD driver. Also includes dual SPI or I²C interfaces which can be used simultaneously as well as internal Data EEPROM.

### STD Flash MCU HT66/67/68Fxx Series

- Wide Operating Voltage Range: 2.2V~5.5V
- Multiple Low-power Operating Modes
- Fast Wake-up Function
- Industrial Specifications: -40ºC ~ +85ºC
- Multiple External Communication Interfaces
- Internal Accurate High/Low Speed Oscillators
- High Noise Immunity and Excellent ESD protection


| Touch Flash MCU | STD Flash MCU | STD 8051 Flash MCU | USB STD Flash MCU | 32bit MCU | Enhanced OTP MCU |
|---|---|---|---|---|---|
| TinyPower™ MCU | Power Management | UART MCU | Phone MCU | EEPROM | WLED Backlight |

HOLTEK SEMICONDUCTOR INC.

*Holtek Semiconductor (USA), Inc.*
*46729 Fremont Blvd., Fremont, CA 94538 Tel: (510)252-9880 Fax: (510)252-9885*
*E-mail: sales@holtek.com http://www.holtek.com*

Fergus Dixon (Australia)

# Smart Switch Management

## Construct an MCU-Based, 'Net-Enabled Controller

Smart switches are handy energy-saving devices that simplify energy conservation. This microcontroller-based, 'Net-connected controller enables you to manage up to 50 switches with ease.

"Mate, we have a new project for you. You'll like this one," said my pal from the contract assembly company. New projects are often referred to contract assembly companies and PCB designers, so it pays to be on good terms with them. This project was to design a controller for up to 50 smart switches. Smart switches are energy-saving devices installed in office blocks to automatically turn off the lights at the end of the day to conserve energy. The controller needed an accurate real-time clock (RTC) that would pulse a 24-V AC line once or twice to turn off the smart switches at the end of the working day, and repeat at two-to-three-hour intervals in case the lights were turned on. I added the Ethernet interface after the first prototype was finished.

## PROTOTYPE DESIGN

The requirements from the customer included a table page of logic, a build price of no more than $100, and a diagram (see Figure 1). The project looked straightforward and could be designed using an Atmel ATmega32 microcontroller, an LCD, an RTC, switches, and a toroidal transformer to supply the power for the smart switches. The aim was to build the project with a minimal amount of problems—or "gotchas."

For the display, a two-line character LCD was a good choice with a simple 8-bit interface and, more importantly, the ability to display 32 characters to show the time, date, and other messages. The LCD backlight adds a professional look. When using Hitachi HD44780-based displays, it helps to use the read/write line to check the busy bit before writing to the display, otherwise the display may go blank if the timing between commands is too short.

For the RTC, I used a Maxim Integrated Products DS1302 trickle-charge timekeeping chip. There are other DS13*xx* clock chips with $I^2C$ and SPI, but since I had used the inexpensive DS1302 before, I used it again for historical reasons. A back-up battery keeps the DS1302 clock running when the power is off, and a CR2032 coin-cell battery will last for more than 10 years. One issue with the DS1302 is the specified crystal is not the same as the standard 32-kHz crystals from the local electronics store. (The specified crystal for the DS1302 should be a 32-kHz, 6-pF type, not the more common 32-kHz, 12-pF type.) The standard crystals will

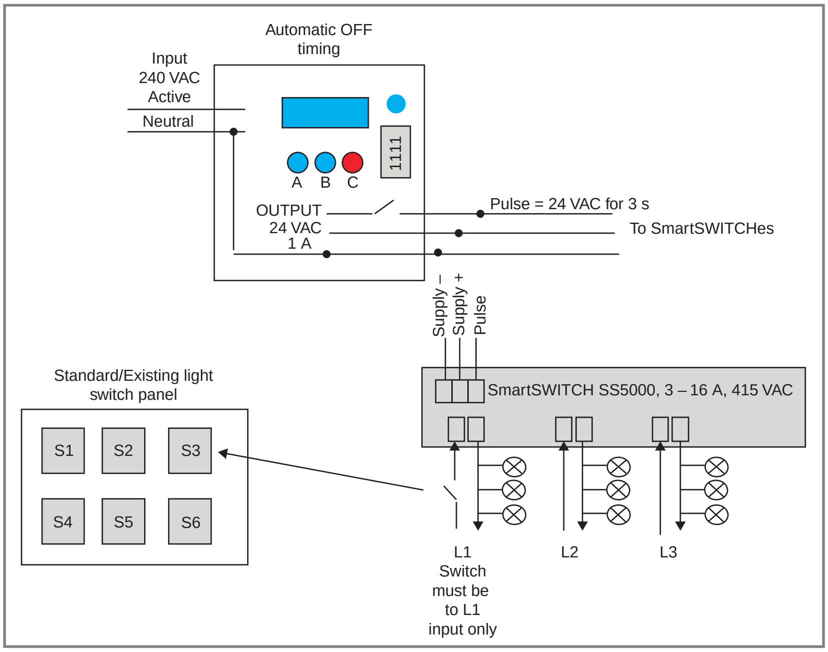



Figure 1—Customer specification diagram

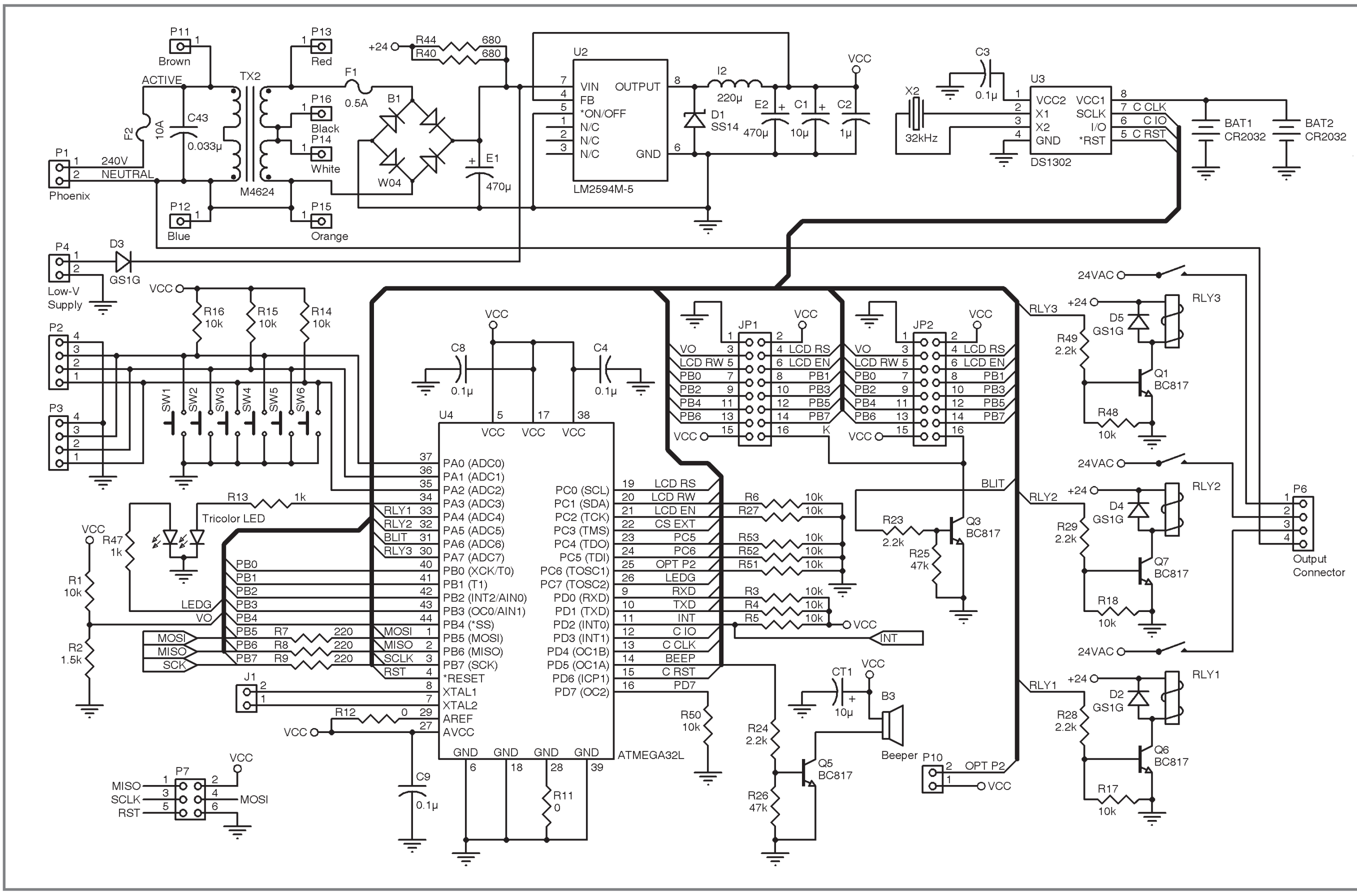



Figure 2—The ATmega32-based controller includes a National Semiconductor LM2594 regulator and a Maxim DS1302 chip.

work, but the clock will be a minute off each day while the specified crystal will keep accuracy to 1 s per day. I'd discovered another issue during a previous project. When the crystal was placed close to an LCD backplane pin (within 2 mm), the crystal preferred to oscillate at the LCD backplane frequency, which was 10 times slower than the 32 kHz required.

The rest of the hardware was simple: two relays to control the smart switches and a 24-to-5-V National Semiconductor LM2594 step-down switching regulator to drive the LCD backlight. I took the schematics and PCB design from other projects (see Figure 2 and Figure 3). Having a library of previously used footprints meant a low risk of PCB layout problems. The PCB design process summarized is to design the schematic, run the electrical rules check (ERC), fix any ERC errors, make a Netlist, draw the PCB board outline, load the Netlist, route the tracks, perform a design rules check (DRC), fix any DRC violations, and then perform an overall check. The overall check is highly recommended, as is double- and triple-checking everything and, if possible, obtaining a peer review from a colleague. Always send the customer a picture of the layout so the option is there to "share the blame" if there are any problems.

The first prototype arrived after a few weeks and was soldered up in-house. With prototypes, there is usually at least one issue, and this time it was that the toroidal transformer outline was drawn on Mechanical Layer 1 not the Keep Out layer. This

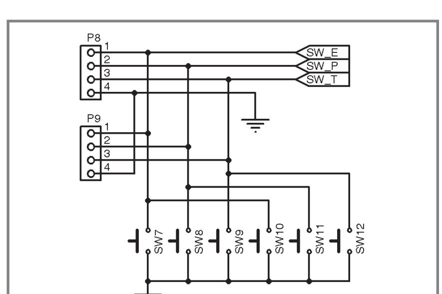



Figure 3—Another view of the controller schematic

meant an unexpected large hole in the middle of the board. Developing the program was straightforward. I cut and pasted code from previous projects and used some custom logic to glue it all together.

## ADDING ETHERNET

It was looking good. Then the customer asked: "The controller really needs to have network access. How hard would it be to add an Ethernet port?" This was new. Revving up the Google engine revealed the Serial Line Internet Protocol (SLIP), which sounded promising, and serial ports were easy. After trying to talk the customer into this, it became obvious that this was not the way to go. Also, the SLIP had the feel of old technology and didn't seem that easy to implement. Another option was a Microchip Technology ENC28J60 standalone Ethernet controller that interfaces using SPI at a cost of $6 for a single piece from Farnell. Even better, it came in two packages, a through-hole 28-pin skinny DIP

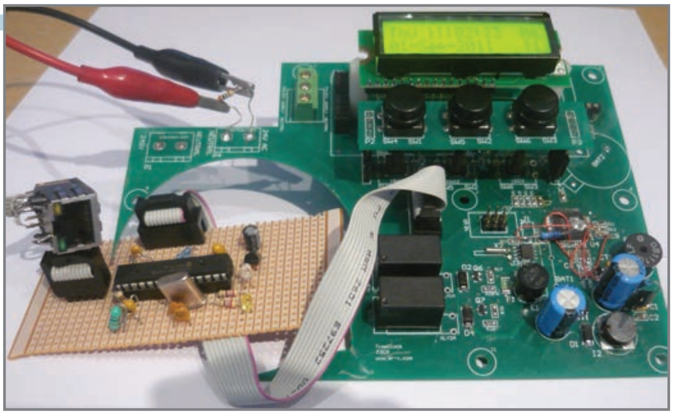


Photo 1—The first prototype featured a Microchip Technology ENC28J60 Ethernet chip on a Vero board.

(good for prototyping) or a 28-pin SOIC package (great for production). The ENC28J60 uses a 3.3-V power supply but has 5-V tolerant inputs, so an extra 3.3-V regulator was required.

This question about adding Ethernet was posted on the AVR Freaks forum, and seemed to be a hot topic with a dozen answers in a short time. Newsgroups, forums, and message boards are great places to visit when unsolvable problems crop up. There is nothing like a crowd-sourced solution. The first reply came back, "You still need a TCP/IP stack."

There are Ethernet modules that can connect to embedded systems. Lantronix adapters are commonly used and, in quantity, the price is reasonable. But it still would have been the most expensive part on the board, so the ENC28J60 part won out here based on its price.

Ethernet connectors have eight pins, which are used for four twisted pairs. For standard Ethernet 10/100, only two twisted pairs are needed that are available on pins 1, 2, 3, and 6—very similar to RS-422. There must be a reason why pin 6 was used instead of pin 4, maybe another historical reason. So, to add Ethernet, the hardware was taken care of with the ENC28J60, but the program would need to understand TCP/IP. I was off to the bookstore, and one TCP/IP book later, the whole project was no clearer. Then, Google popped up a webpage of a project showing a web server project using an Atmel ATmega88 microcontroller.[1] This page included a schematic, a walkthrough of the steps to implement the TCP/IP, and the source code, which was available for download.

I purchased a single ENC28J60 chip, and an add-on board was built using a Vero board (see Photo 1). A special connector from Bel Stewart Connector called a MagJack was used for the eight-way Ethernet jack. MagJacks have two internal isolating transformers and two LEDs (yellow and green) and are slightly larger than a standard RJ-45 eight-way jack. The only downside to the MagJack sourced from Farnell was the price of $18. After doing some simple tests to make sure the SPI port was writing to the chip, the TCP/IP source code was cut and pasted into the program and the original web server project Output on/off webpage was displayed.

The HTML code of the original project was modified to suit this project, which was now called the Webclock. Some text boxes and buttons were added for the settings, and the program was altered to beep and change the single tri-color LED to orange for 2 minutes when the Web button was pressed. Pressing a button on a webpage activates the Post method, which takes the browser to a new page, so a new IP address setting was added to enable any button press to reload the same page (see Figure 4). The IP address is a series of four numbers separated by a period, with each number ranging from 0 to 255. An IP address of 10.1.1.15 was used for testing. This is referred to as Internet protocol version 4 (IPv4). IPv6 is a new system currently being rolled out, which will eventually replace IPv4.

In the imported code, there was a place where the Ping command would be processed. On a computer, typing in the command "Ping 10.1.1.15" would show the delay in milliseconds for three replies from the controller. In the ATmega32, the best way to send out the HTML page was to have a large string array in SRAM, which was loaded up with the values and sent out all at once. This reduced the delay. The ATmega32 had 2-KB RAM, which was enough—75% of this was devoted to the HTML buffer. During this stage, where the HTML and the edges of the TCP/IP code were being modified, a process was needed to actually check what was being sent out. I found a network protocol analyzer called Wireshark, which is a free download. It is easy to install and run, but the amount of data shown can be overwhelming. A filter was set (ip.addr==10.1.1.15) to only view the packets sent to or received from IP address 10.1.1.15. This showed the Get packet which redrew the webpage and the Post packet activated from webpage button presses. Several hours were spent using Wireshark to debug the interface, but the amount of time saved was far more.

The PCB was redesigned and a second prototype was made. A potted toroidal transformer from Altronics was tested under its full 1-A load and the temperature rise was 30°C, which was deemed acceptable. After finalizing the program and doing some thorough documented tests, this second unit was dispatched.

## ACCESSING THE INTERNET

A few days later, the customer called to say the controller worked well and was now connected to the Internet. It could be accessed by typing an IP address into a web browser (http://121.223.229.168/esco), but the buttons would not work via the Internet. This was unexpected but very interesting. When modems connect to the Internet, they are usually assigned a dynamic IP address that appears random, but the customer explained he had a static IP address from his ISP. A static IP address is an IP address

```
<FORM action="http://10.1.1.15/password"
method="POST">New Time <INPUT type="text"
name="SetTime"> <INPUT type="submit"
value="Set"><BR>
```

Figure 4—HTML code for a webpage button

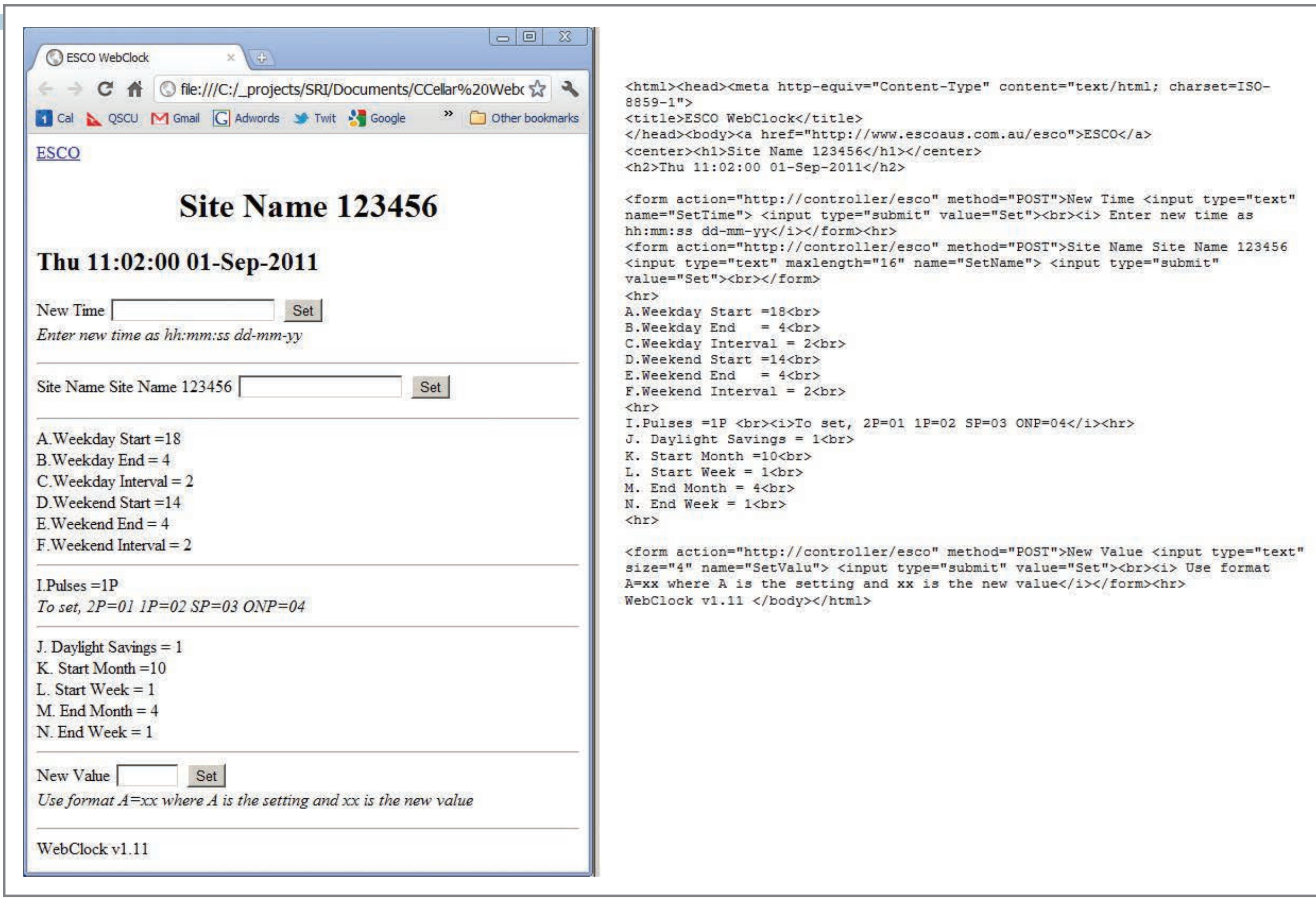



Photo 2—Webpage and HTML page source

that is always the same four numbers every time the modem logs onto the Internet. The customer had then set up port forwarding on his modem (Siemens SpeedStream 6520), which enabled the controller to be accessed externally.

A quick look at the HTML code sent out by the controller showed the buttons were still set to load the local IP address of the host computer (see Photo 2). The web browser was trying to access a local page on the customer's computer, which was not available through the Internet. I did a quick edit of the HTML code using Notepad to set the button action to load the static IP address, and then the button action worked and the settings could be changed remotely.

This was great: Ethernet was working and access through the Internet was working. What could possibly go wrong now? The gotchas were all tied up and disposed of. The only small issue left was how to access the controller locally and

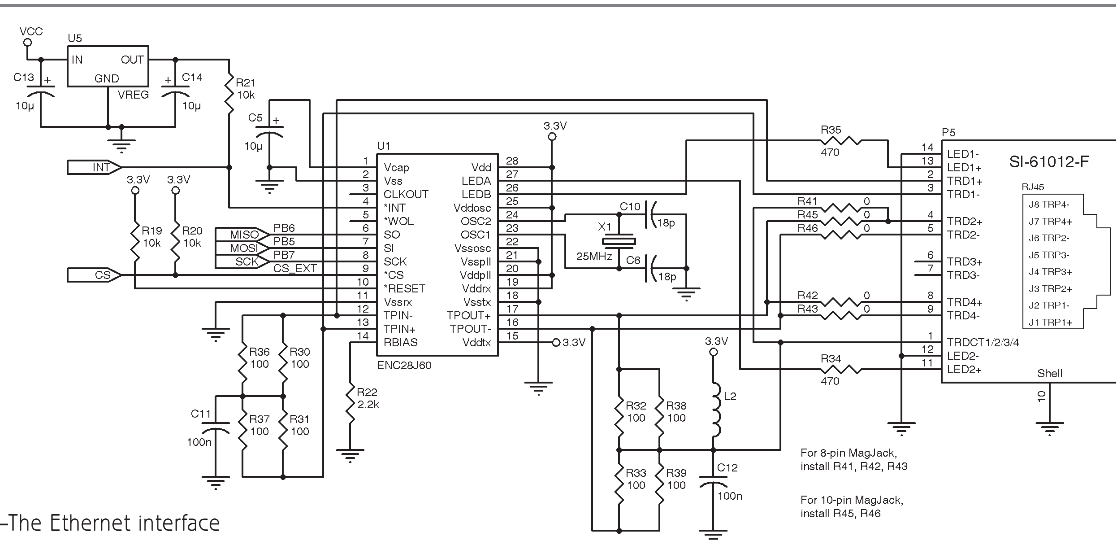



Figure 5—The Ethernet interface

remotely through the Internet. Easy, just tell the customer it wasn't possible. There was some disagreement here, and it was back to the drawing board. There is a saying that the last 10% of the project always takes the other 90% of the time.

### THIRD PROTOTYPE

While this local-versus-remote problem was being investigated, some changes were proposed to the hardware. The two most expensive components were the 10-pin MagJack and the potted toroidal transformer. Sparkfun Electronics had similar RJ-45 Ethernet MagJack-compatible connectors in stock for $1.56. These connectors had eight pins, so the PCB was modified to be able to take both connectors and five resistors were used to select the eight- or 10-pin MagJack (see Figure 5). The yellow and green LEDs on the Sparkfun MagJack were back-to-front compared to the MagJack from Farnell, but this was easily fixed in the program. The green LED was set to turn on when the Ethernet port was active and the yellow LED was set to blink when packets were received or transmitted. The ENC28J60 part has a register to set up the LED configuration. With the toroidal transformer, the assembly company had found an open toroidal transformer for a quarter of the price. Spade connectors were added so the transformer wires could be crimped and plugged on. Doing the full 1-A load test for 1 hour, the temperature rise was now 8°C, compared to the previous 30°C rise for the potted transformer. The bill of materials without the PCB, assembly, or test costs came to just over $96.

The entire PCB was designed to be fitted into a wall-mounted PacTec box. Due to the depth of the box, the switches had a separate board, and a long-tail header was used for the LCD (see Photo 3). The front of the box had a recess for a Lexan label, so the next step was to design this label in Adobe Illustrator. Graphics companies only accept Adobe Illustrator files (or sometimes Corel Draw) since these graphic files are vector based—the pictures can be scaled to size without losing resolution and becoming pixelated. The PCB outline, the LCD, and the three switches had their outlines copied to Mechanical Layer 3, and the PCB file was exported as a DXF file. This file was then imported into Adobe Illustrator and all layers, except for Mechanical Layer 3, were deleted. The customer sent the artwork for their logo, which was also imported, and a short time later, a first draft of the label was ready for quotes.

The program was modified to add some minor changes. One change was to add a separate finish time for the IT department since IT personnel start earlier and finish after the rest of the office. The clock needed to negotiate Daylight Saving Time, and this meant jumping the clock an hour forward at the start of summer and then an hour back six months later. This was not trivial, and a bug was found. Coming out of Daylight Saving Time, the clock was set back an hour from 3:00 A.M. to 2:00 A.M., which then put the clock back into Daylight Saving so the clock was set

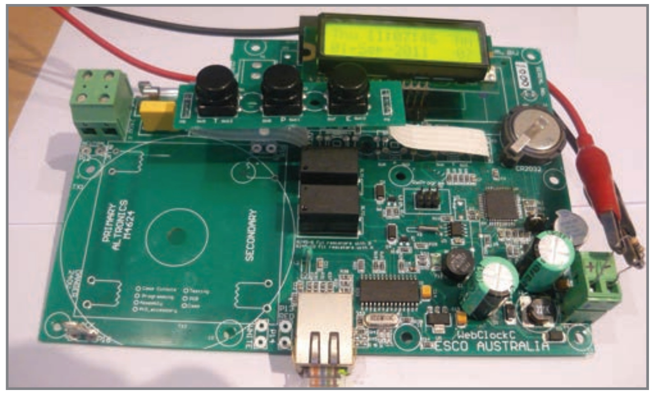


Photo 3—Final PCB with a surface-mount Microchip Technology ENC28J60 Ethernet chip

forward an hour to 3:00 A.M., which took the clock out of daylight saving, and so forth.

## REMOTE VS. LOCAL CONNECTION

The customer suggested adding a port to the IP address to fix the remote/local access issue. The Internet port number is a number that is added to the IP address to describe the device or action of the device. All Internet HTTP connections use port number 80, while mail uses port number 25. The IP address should actually be written as 10.1.1.15:80, where 80 is the port number. But if the port number is missing, it is assumed to be 80. Unfortunately, changing the port number was not the solution here.

To replicate the customer's setup, I set up port forwarding on our D-Link DSL-G602T modem. All modems are not the same, and after a few visits to a local computer service agent, it became apparent the D-Link modem had the port forwarding feature, but it wasn't going to work. It was easier to use the same model of modem as the customer's. A quick scour of eBay and a second-hand unit was on the way.

This problem was proving pretty tough, and several months had now passed with not much progress. At times like this, it helps to outsource and ask colleagues. LinkedIn had a few connections to IT personnel from previous jobs. After a few e-mails back and forth, the answer arrived. There is a text file named Hosts that sits in the Windows/System32/Drivers/etc directory. Inside this file, it is possible to define an IP address lookup for a name. For example, localhost may be 127.0.0.1 and more importantly, Controller can be defined to be 10.1.1.15. This enables the host's file on a local and remote computer to be set to:

```
Local Computer    Controller   10.1.1.15
Remote Computer   Controller   Static IP
     address which was 121.223.229.168
```

This naming system is the basis of the domain name system (DNS), where domain names are converted to IP addresses. A new setting was added to the program to enable a word to be adjusted and the access setting could be set to local, external, or DNS.

The customer was a little underwhelmed, but was convinced after some discussion. This was a solution and provided a finish for the entire project, even if it did not at first seem elegant. The setup to edit the host's file on the local and remote computer was worth it. From start to this point, the project had taken six months.

## THE FINAL PROBLEM

The only other issue was that when an Internet USB Wi-Fi adaptor was plugged in, the controller did not easily talk to the local computer. No one else had this problem, so it was shelved. Later on, while trying to set up a Secret Labs Netduino Plus development board, the same problem reappeared, and it was fixed by turning off the Ethernet setting: "Obtain an IP address automatically."

So, while this project did not always go smoothly, I overcame the roadblocks by bringing in experts from the outside. Newsgroups and forums are a great way to ask questions, especially late at night when everyone else is asleep. For quick answers, e-mail or phone a friend. The sample web server project was the key to completing the Ethernet interface.

*Fergus Dixon (fdixon@sr-i.com) holds a BE from Sydney University. After working for 10 years in various fields such as packaging, mining, and medical and control systems, he set up his own company, SRI, to do custom electronic designs. After 14 years (and two Australian design awards), Fergus now runs several businesses including Electronic System Design and Shields for Arduino. He recently designed the popular program, "Simulator for Arduino."*

## PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/263.

## REFERENCE


[1] tuxgraphics.org, "HTTP/TCP with an ATmega88 microcontroller (AVR Web Server)," www.tuxgraphics.org/electronics/200611/embedded-webserver.shtml#0lfindex3.


## RESOURCES

AVR Freaks Forum, www.avrfreaks.net.

Wireshark, www.wireshark.org.

## SOURCES

**12 + 12 PCB 30-VA Toroidal transformer**
Altronics | www.altronics.com.au

**ATmega32 and ATmega88 Microcontrollers**
Atmel Corp. | www.atmel.com

**MagJack Connectors**
Bel Stewart Connector | www.stewartconnector.com
Farnell (Supplier) | www.farnell.com

**HD44780 Dot-matrix LCD controller/driver**
Hitachi Ltd. | www.hitachi.com

**DS1302 Trickle-charge timekeeping chip**
Maxim Integrated Products, Inc. (Dallas Semiconductor) | www.maxim-ic.com

**ENC28J60 Standalone Ethernet controller with SPI**
Microchip Technology, Inc. | www.microchip.com

**LM2594 SIMPLE SWITCHER power converter 150-kHz 0.5-A step-down voltage regulator**
National Semiconductor Corp. | www.national.com

**Netduino Plus development board**
Secret Labs | www.netduino.com

**RJ-45 Ethernet MagJack-compatible connectors**
Sparkfun Electronics | www.sparkfun.com

Miguel Sánchez (Spain)

# Image Processing System Development

## Use an MCU to Unleash the Power of Depth Cameras

You can accomplish a variety of computer vision-related tasks with a microcontroller, bipolar stepper motors, and a depth camera. This article details how to use embedded technology with Microsoft's Kinect motion-sensing device for innovative image-processing applications.

There is no doubt that Microsoft's Kinect motion-sensing device has been a huge commercial success. What is a bit less known is the story of how people have been hacking the device for a variety of purposes.

For those of you who've been living in a cave for the past few years, Kinect is a peripheral device for Microsoft's Xbox 360 game console (see Photo 1). It senses user motion for a hands-free gaming experience. You move your arms and legs in front of a TV set and your body gestures are translated to your character in the game. It could be said that Kinect is a camera, but it is much more than that.

*Circuit Cellar* is not a gaming magazine, so you may be wondering what this article is all about. Let me start from the beginning.

### ART PROJECT

I recently worked with Rubén Tortosa—local artist, Fine Arts professor, and friend—to obtain the silhouettes of the visitors of a given installation for his art exhibit titled "TRAS (la espera)." This silhouette would later be printed on one of the exhibit's walls. Photo 2 shows the wall being printed. The initial idea was to use a computer vision algorithm called background subtraction. If you set a camera facing an empty wall, a frame with the empty wall scene can be captured and it later can be subtracted from the next frames captured. If a person is then in front of the camera on a new frame, the difference between these two frames will be noticeable across all the pixels that change brightness and color values between the two frames. Many of them will be the body of the person standing in front of the camera.

Of course, a few things can be done to improve the accuracy of the detection and to compensate for changes in the scene lighting. Shadows are one thing that causes problems with this approach.

Testing this idea was straightforward. With my laptop's webcam and a few lines of code, I could capture an image, store it, and compare it with another frame using Processing, which is an open-source development environment based on Java that can be used to create media-rich applications. It was developed at MIT and is freely available for Windows, Linux, and OS X. During the last two years, Processing has been my preferred choice for prototyping.

Photo 1—A structured light pattern comes from the Kinect's leftmost "eye."


Photo 2—The exhibit featuring the finished project created quite a stir with the public. Here you can see a front view of the wall being "printed."

## THE KINECT HACK

While I was implementing the aforementioned approach, an unexpected turn of events occurred. Kinect hacking started to get news coverage.

As I mentioned, Kinect was originally presented on the news as a new gaming peripheral for a game console. Nothing in the marketing campaign suggested Kinect could be used for any other purpose. However, Kinect uses USB to connect to the Xbox 360 game console.

Given the features that were apparent from the available games, many people started to think this little device could have many interesting uses beyond gaming. There was a problem though: neither drivers nor documentation were available. And then some people decided that it might be possible to reverse-engineer the protocol Kinect uses over USB. There are hardware-based USB sniffers that can log the two-way communication between an Xbox 360 and a Kinect device.

MIT-trained engineer and entrepreneur Limor Fried of Adafruit Industries bought a Kinect and obtained a trace of the USB communication. But, instead of sitting there trying to figure out how it worked, she took the 2.0 approach and crowd-sourced the research. Her company offered a $3,000 bounty to the first person who figured out how the dialog worked based on the communication logs she provided on the company website. The wait was not long. A few hours later a Spanish student, Hector Martín, had a winning entry.

Initially, Microsoft didn't react well to Kinect hacking. There were even rumors of lawsuits. But since then, the waters have calmed and an SDK for Windows 7 is now available. This means Microsoft no longer has a problem with people using Kinect for purposes other than playing with Xbox.

As a result of the hacking effort, an unofficial open-source driver became available and an interesting set of details about the device and its capabilities became public. Weeks later, PrimeSense, the company that licensed the optical technology used by Kinect to Microsoft and other partners, released an open-source official driver called OpenNI.

## THE DEVICE

As you can see in Photo 1, the Kinect has coin-size circles on its plastic frame. One is just a regular webcam. Behind the other two is where the magic happens. Kinect utilizes structured light using an infrared laser and an infrared camera to capture the reflection of that light pattern. Processing of the latter image determines the depth of each pixel of the infrared image.

In the depth image, each pixel value represents the distance from the Kinect to it. That means a 3-D representation of the scene is obtained, where each pixel has an X,Y coordinate plus the depth value, which is the Z coordinate.

The distances measured range from 30" to 18'. Anything closer or farther away is represented as zero distance. Kinect has a tilt motor to make user tracking easier. You can command it to raise or lower the device attitude so the user remains completely inside the scene. It also has a three-axis accelerometer that detects any motion of the device. Audio commands are possible as the device also has four microphones. Finally, you can switch one front LED on or off. It can flash in two colors in response to commands from your computer.

## THE LIBFREENECT LIBRARY

Due to the effort of volunteers for the OpenKinect Project, a library and bindings for different programming languages are available. As it is the result of hacking the USB

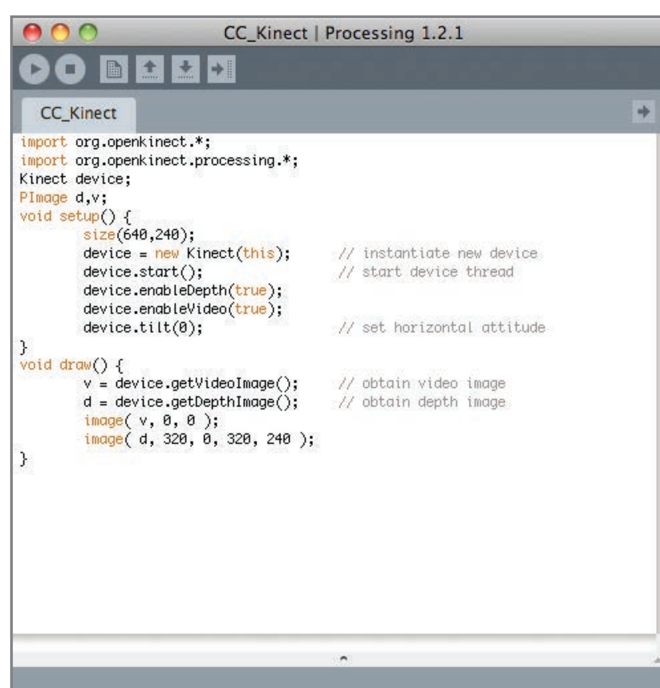



Photo 3—This sample Processing program shows both types of Kinect images: RGB video and depth image. Notice how similar this is to the Arduino integrated development environment.

communication, certain features, like the audio, aren't yet supported.

Once detected, the device can grab color frames, infrared frames, or depth-info frames. Front LEDs can also be controlled and datastream from accelerometers can also be read.

While most of the library code is developed in C++, the library itself provides a broad selection of wrappers so you can access it from different programming languages including C++, Java, Python, ActionScript, and C#. And some of the provided wrapper code, as the ActionScript server, enables any other language with TCP/IP sockets support to interface with the library.

Depending on the user platform, ready-to-use packages may be available. The first thing I tested was an example developed for OS X that used the openFrameworks C++ toolkit. Unfortunately, this first example required me to download the latest version of Apple's Xcode toolchain, which at 3.5 GB took a while to download.

There is now a repository for binary versions for Ubuntu and several choices for OS X. Binary versions make user life easier, but full source code is available for download from GitHub. If you have to compile from source, remember to honor the dependencies (mostly libusb-1.0). Together with the library comes the program glview that will enable you to double check whether or not the device is accessible in your system.

One of the easiest ways to use the library is with Processing (see Photo 3). Processing code is based on Java, but it makes a programmer's life much easier than just using Java when dealing with graphics, video, sound, and serial communications. Daniel Shiffman, an Assistant Arts Professor at NYU's Tisch School of the Arts, created a library to use libfreenect for Processing language for OS X. Later, other contributors figured out how to use it for other platforms including GNU/Linux.


Photo 4—The sample program's output: RGB video (a) and the band depth image (b). Note some of the black shade behind the foreground body. In the depth image, brighter means closer.

Processing is an easy environment with lots of sample programs and online tutorials. And the library code comes with some examples, too. So once you get it installed, you can start fiddling with the code from the examples. To get everything working, you need to build a libfreenect library and the Java library and to get the latter copied to the libraries folder of your Processing working folder. This might be tricky because there is room for error. Once you get the example that comes with the library running, you know everything is properly set up.

You can connect several Kinect devices to the same computer. A program starts by instantiating a Kinect object that needs to be started. You can enable depth images and color images individually. Then, the program can choose a new depth or color frame by calling the appropriated function.

The sample processing program produces an output showing the regular RGB video image and the depth image as a monochrome image (see Photo 4). Some other software represents the depth image as a color image where different colors represent different distances. This is similar to the way thermal images are sometimes represented. In any case, the truth is that distance has no color or grayscale value, but it comes in handy to have a visual way of showing the depth value.

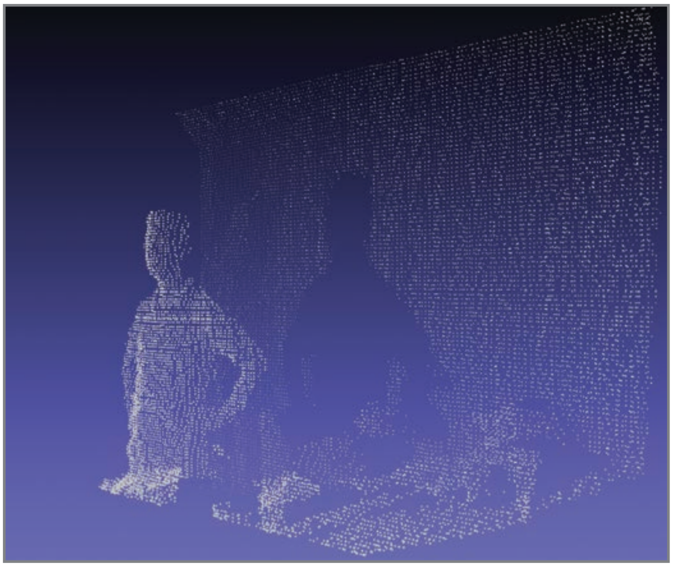


Photo 5—Here is a sample depth image projected in a 3-D space. Note how the body in the front hides what is behind it.

## THE DEPTH IMAGE

Most of the Kinect's beauty comes from its ability to offer us a depth image. In Photo 5, each pixel represents the distance measurement from a Kinect device to that particular object. The pixel's value is not connected to the object's color. One simple way to represent that distance measurement is by using a grayscale conversion: pixels whose values are higher are represented by a light gray (or almost white), while darker shades of gray represent lower values (see Figure 1). The

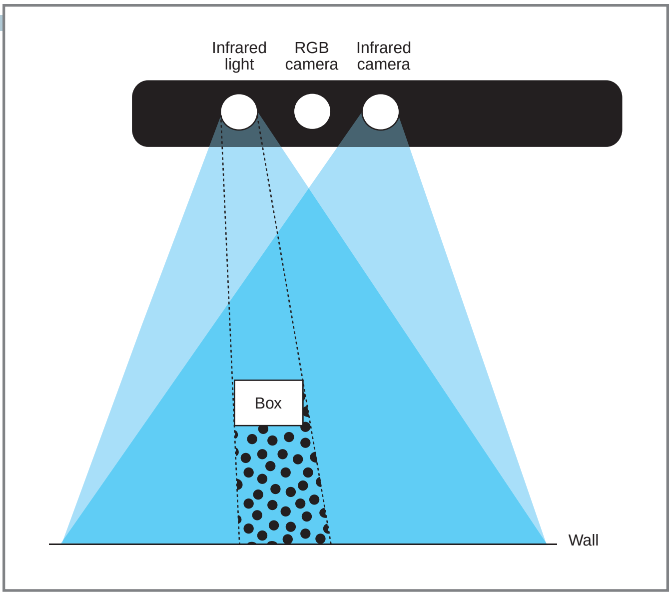



Figure 1—Some areas of the depth image are black, as they are in the shade of the infrared illumination, and therefore invisible from this angle. The dotted area behind the box is invisible to the depth camera, and it is rendered as a black shadow.

Kinect depth image can be misleading because some pixels do not convey information due to the shadows.

Alternatively, it is possible to transform the distance value into different colors instead of grayscale values. This way, a wider range of numbers can be represented and better distinguished by the user.

A depth map of a scene enables the software to determine the distance to different objects in the scene, which is not an easy task using computer vision algorithms. A depth image can filter out areas of the image that are too close or too far away to be relevant. It's much easier to obtain human gestures or extract silhouettes from a depth image than it is from a regular picture. In addition, the results are more reliable and independent of the scene's illumination. The only limitation is that direct sunlight on scene objects may blind the Kinect, because isn't designed for outdoor use. Please note that a little transformation is needed to obtain the proper distance (in meters) for a raw measured depth value. Distance in meters is:

$$\frac{1}{(-0.00330711016 \times raw\_depth + 3.3309)}$$

With a depth image, the programmer can create an XYZ cloud with a simple transformation. Each X,Y coordinate of each pixel translates to a 3-D space, where depth value represents the Z coordinate and X and Y are obtained by projecting X,Y values:

$$X = \frac{(X - 320) \times Z}{600}$$

and

$$Y = \frac{(Y - 240) \times Z}{600}$$

You can adjust all the constants in these equations to carefully represent device calibration. You can see more details in the PointCloud example code that comes with the Kinect library for Processing.

## PROJECT HARDWARE

The project consists of a PC running Ubuntu 10.4, a Kinect camera, a motor controller board built around an Arduino (featuring an Atmel ATmega168 microcontroller), and two bipolar stepper motors with enough torque to hold the weight of the pen holder and the belts. I had some stepper controller boards around my desktop built around Allegro MicroSystems's A4988 DMOS microstepping driver that offer up to $16^{th}$ step microstepping that could handle up to 2 A per coil, which was enough for my steppers. I wanted to be able to experiment with different microstepping resolutions, so I made that selection software configurable from the Arduino. Each motor's motion is controlled by two digital inputs, one for the stepping clock and one for selecting step direction. A RESET pin on the controller resets the internal logic and disables the output drive current. I wired my circuit on a protoboard (see Figure 2). But it later

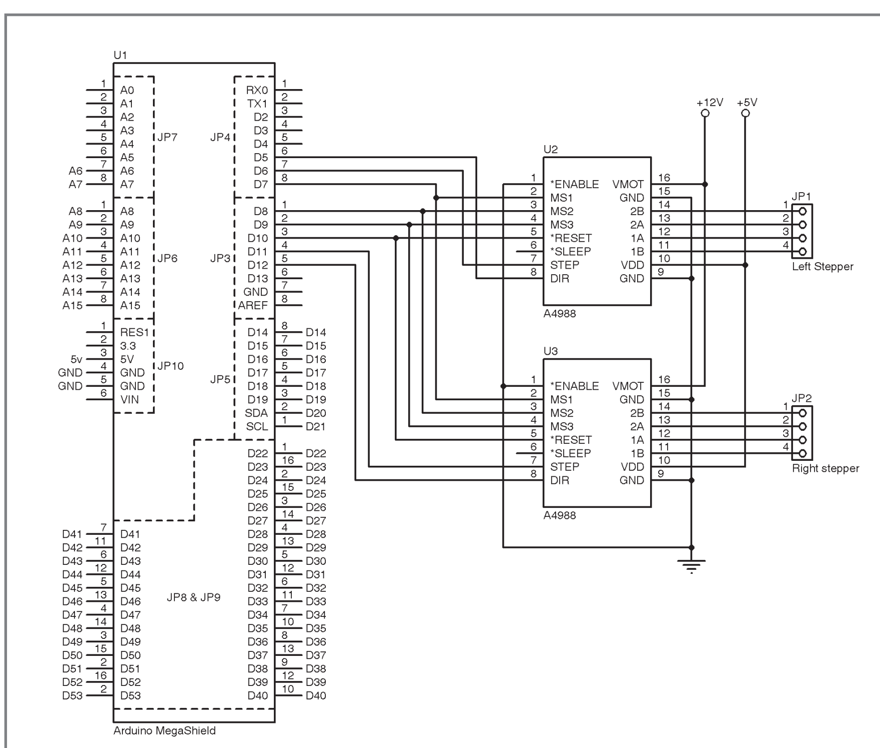



Figure 2—The driver board features an Arduino Mega. But you can use an Arduino UNO without a problem.

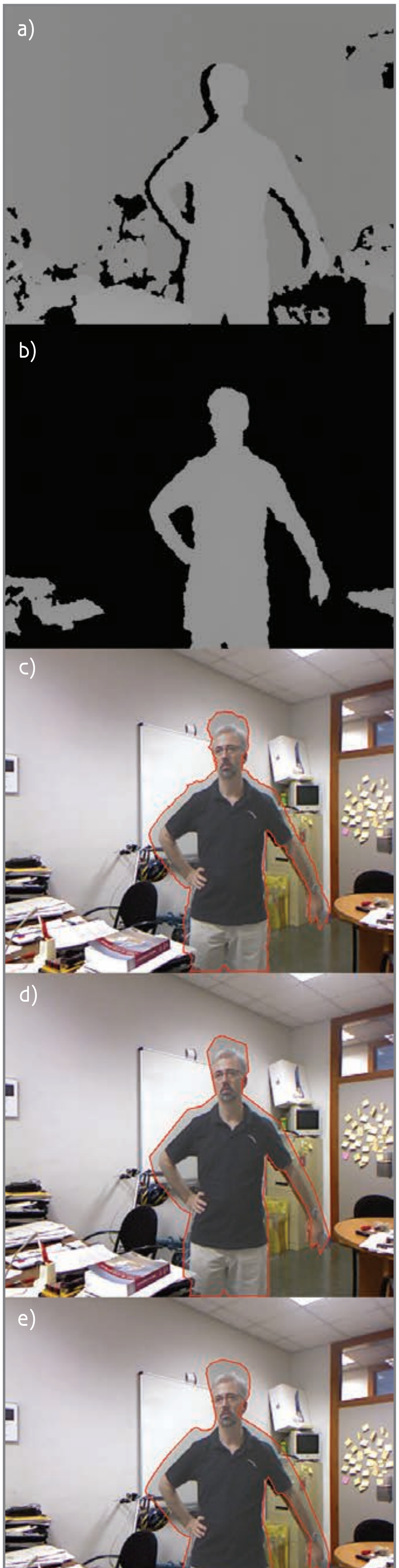



Photo 6—These are the five steps to image processing: depth image (a), umbralized depth image (b), contour extraction (c), simplified contour (d), and spline-smoothed contour (e).

occurred to me that a RAMPS board from a RepRap project could be used instead, as it interconnected an Arduino Mega board with up to five stepper motor controllers and an inexpensive PCB can be found on eBay. If you go this route, please note that mode selector pins MS1, MS2, and MS3 would then be jumper-selectable only.

## PROJECT CODE

My project started as a simple enterprise that later became a bit more challenging. The idea of capturing the silhouette of an individual standing in front of the Kinect was based on isolating the points between two distance thresholds from the camera. As a depth image already provides the distance measurement, all the pixels of the subject will be within a range of distances, while other objects in the scene will be outside of this small range. But I wanted to have just the contour line of a person and not all the pixels that belong to that person's body. OpenCV is a powerful computer vision library. I used it because of the `blobs()` function, which extracts the contour of the different isolated objects in a scene. As my image would only contain one object—the person standing in front of the camera—the `blobs()` function would return the exact list of coordinates of the contour of the person, which was what I needed. Please note that this function is heavy image processing made easy for the user. It provides not just one, but a list of all the different objects that have been detected in the image. It can specify whether holes inside a blob are permitted. It can also specify the minimum and maximum areas of detected blobs. But for my project, I am only interested in detecting the biggest blob returned, which will be the one with index zero, as they are stored in decreasing order of blob area in the array returned by the `blobs()` function.

Though it is not a fault of the `blobs()` function, I quickly realized I was getting more detail than I needed and that there was noise in the contour's edges. Filtering out on a bitmap can be easily accomplished with a blur function, but smoothing out a contour did not seem as obvious to me.

Removing certain points can simplify a contour line. A clever algorithm can do this by removing points close enough to the overall contour line. An example is the Douglas-Peucker recursive contour simplification algorithm. The algorithm starts with the two endpoints. It accepts one point in between whose orthogonal distance from the line connecting the two first points is larger than a given threshold. Only the point with the longest distance is selected (or none if the threshold is not met). The process is repeated recursively, as new points are added, to create the list of accepted points. (Those that are contributing the most to the general contour given a user-provided threshold.) The larger the threshold, the rougher the resulting contour.

By simplifying a contour, human silhouettes look better and noise disappears, but they look a bit synthetic. The last step was to perform a cubic-spline interpolation so the contour became a set of curves between the different original points of the simplified contour. It seems a bit twisted to simplify first and later add back more points because of the spline interpolation. But this way it creates a more visually pleasant and curvy result, which was my goal. Photo 6 shows the entire process, where an offset between the human figure and the drawn silhouette is apparent. The offset is due to how the Kinect works. On one hand, the depth image is obtained from the infrared camera, which is different from the RGB camera that takes the picture. As a result, each camera may have a slight difference in its alignment and optics. On the other hand, they are sitting side-by-side in the device, so they do not capture the same image.

You can manually calibrate both cameras so a colored 3-D point cloud is created (see Photo 7). However, I didn't need this for my project because only the silhouette was required. But proper calibration code is needed for some applications that require a rotation matrix and a translation vector. An interesting project by Nicolas Burrus, a postdoctoral student at University Carlos III of Madrid, enables you to calibrate your Kinect using a chessboard printed on a sheet of paper.

The resulting human silhouette will later be painted on a wall at the art exhibit using a larger version of the vertical plotter featured in my 2008 article, "Vertical Plotter System" (*Circuit Cellar* 212). I used an Arduino board to control the steppers motors. Processing-language code handles the entire system and sends motion commands to the Arduino board.

## ARDUINO CODE

From the very beginning, my plan was to reduce the microcontroller code to a bare minimum. Although it is possible to do all the vertical plotter's geometric calculations within the Arduino board, it would be complicated to adapt the installation for different wall sizes. And it would be more complex because firmware had to be changed to

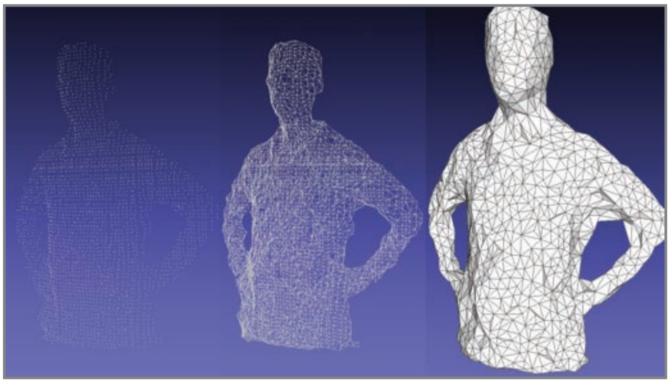


Photo 7—These are steps in the process of converting a point cloud into a 3-D object of a person.

reflect the new installation measurements (the distance between the stepper motors and the belt length at the beginning). Alternatively, it would be possible to preset these values into the Arduino's EEPROM from the computer, but then complex communication between the computer and Arduino board would be needed to update these values when needed.

The final version enables the computer to send four simple commands to the Arduino board controlling the motors: up and down on left and right stepper motors in one step. The installation doesn't include end-stop sensors, because you start it from a known initial location. No messages are sourced from the Arduino board. All the communication comes from the PC through the USB serial adapter. Flow control is not used in the form of any of the four characters "AaBb," where A represents the left stepper motor and B represents the right motor. A lowercase letter means up (stepper motor will wind up one step of the belt) and an uppercase letter means down. I chose a speed of 1,200 bps for the serial interface to prevent a buffer overrun. This way, the time devoted for each step is shorter than a byte transmission time, which ensures the last step ends before a new command arrives.

## THE WONDER OF OPEN SOURCE

While initially aimed at the gaming market, the Kinect has proven to be an interesting device for many other purposes. Microsoft has been quick to recognize this reality by deactivating lawsuit threats against people using the Kinect for unexpected purposes. As a result, there are many interesting projects featuring this device. Fashion companies are using it for 3-D body scanning. Students are using it to help robots navigate indoors. It is even being used as a tool for the blind. Once you've got a depth image as a point cloud, you can easily translate it into a polygonal mesh.

As for my art project, it shows that there are many useful open-source tools at your disposal. When combined, these devices can enable the creation of fully working systems. I built this project on Ubuntu, and it uses Processing, an Arduino, OpenCV, libfreenect, and a few ad-hoc Processing libraries for full-screen display and Kinect handling. All of it done with open-source tools.

*Miguel Sánchez (misan@disca.upv.es) holds BS, MS, and PhD degrees in Computer Science from Universitat Politècnica de Valencia in Spain. He has been teaching computer networking courses at the university since 1989. Miguel's interest in electronics and microprocessors sparked his career in computer science, but his solder is always at hand. His current research deals with 3-D scanning and 3-D printing because he was bitten by the RepRap bug.*

## PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/263.

## RESOURCES


Adafruit Industries, www.adafruit.com.

N. Burrus, "Kinect Calibration," 2011, http://nicolas.burrus.name/index.php/Research/KinectCalibration.

GitHub, http://github.com.

OpenCV, http://opencv.willowgarage.com.

The OpenKinect Project, http://openkinect.org.

PrimeSense Ltd., OpenNI, www.primesense.com/en/openni.

Processing, http://processing.org.

RepRap,org, http://reprap.org.

M. Sánchez, "Vertical Plotter System," *Circuit Cellar* 212, 2008.

D. Shiffman, "Getting Started with Kinect and Processing," 2011, www.shiffman.net/p5/kinect.

ubaa.net, OpenCV Processing and Java Library, http://ubaa.net/shared/processing/opencv.

Ubuntu, www.ubuntu.com.


## SOURCES

**A4988 DMOS Microstepping driver**
Allegro MicroSystems, Inc. | www.allegromicro.com

**Arduino Mega board**
Arduino | http://www.arduino.cc

**ATmega168 Microcontroller**
Atmel Corp. | www.atmel.com

**Kinect**
Xbox, Microsoft Corp. | www.xbox.com

by Bob Japenga (USA)

# Concurrency in Embedded Systems (Part 1)

## An Introduction to Concurrency and Common Pitfalls

Most embedded systems have concurrency requirements that designers and programmers must identify, understand, and manage. This article is the first in a series about the topic. Concurrency is defined and pitfalls are examined.

I work for a small company that focuses on embedded systems and software design. When we started in the late 1980s, we had no reputation outside our previous industries. In order to get work, it was important for us to establish our credibility and to demonstrate our expertise. Here's how we did that during those early years.

After we got through the first level of vetting with a new company (we weren't criminals and at least we talked a good story), our clients would ask us for a proposal to either modify their existing code or to create new code based on the functionality of their old code. After the exchange of a nondisclosure agreement, they would send us their source code to enable us to quote the job. During that process, I would immediately look at their serial drivers because I know this is an area that is usually prone to errors. Most serial drivers are interrupt-driven and thus require knowledge of how to design concurrent threads in embedded systems. In our own experience and in the experience of others, we had seen that there were many possible pitfalls. If we could help our clients solve an existing problem quickly and for no charge, I thought this would both establish our credibility and demonstrate our expertise. So I would briefly look in their code for some of the common flaws we had both created and seen throughout years of dealing with interrupt-driven serial drivers. In almost 50% of the cases, we were able to find a real problem in their code—and it was always caused by concurrency issues. Sometimes they were serious and sometimes they were relatively harmless.

Getting concurrency right is not easy. Over the next several articles, I will address concurrency as it relates to embedded systems. In this article, I define concurrency, list some of the common pitfalls, and look at one of them in particular. I will address other common pitfalls in upcoming articles.

### DEFINING CONCURRENCY

Concurrency takes place any time two or more activities can happen in the same time segment. For example, I can concurrently wash the dishes, watch TV, and listen for a text message on my phone. A phone can concurrently notify you of incoming calls while displaying a TV program.

According to Wikipedia, in computer science,

### HELPFUL DEFINITIONS

**Thread:** Any functionality that can be active during the same time segment. It can be in hardware or software.

**Blocked:** When a software thread relinquishes control of the processor to the operating system.

**Preempt:** When one thread interrupts another.

"concurrency is a property of systems in which several computations are executing simultaneously, and potentially interacting with each other."

Of course, if they don't interact, there are no pitfalls, but they are still concurrent. However, they almost always interact in some way and that is what creates the problems.

| Our Example Design Without Concurrency |
|---|
| Check to see if the user has hit a key |
| Output the new display as a result of the key press (takes about 100 ms) |
| Check to see if something changed in the external world |
| Process the changes from the input from external world and set the output accordingly |
| Log the user input and the output state to our logging device (takes about 50 ms) |
| Rinse and repeat |

Table 1—Steps that single-loop software may take in a design without built-in concurrency

## COMMON CONCURRENCY PITFALLS

Any embedded system that has concurrency built into its design can experience one or more of the following problems: Race Conditions—where the order of execution affects the outcome of a given result. Corrupting of Shared Resources—where shared resources are used by two or more concurrent threads. Logical Complexity Creating Less than Airtight Algorithms—the more things that can affect each other, the more difficult it is to carefully think your design through and make it bullet proof. Deadlock—where functions stop working for no apparent reason. Time-of-Check-to-Time-of-Use (TOCTTOU)—I read the data from an input and make a decision, but by the time I use the data, the data has changed. Priority Inversion—where a lower-power thread locks out a higher-priority thread.

I will cover these conditions over the next several articles. In this article, I'll address priority inversion: what it is and how to avoid it.

## WHAT IS PRIORITY INVERSION?

Given the number of problems concurrency has given me over the years, how have concurrency issues ranked in the history of major software failures? As I looked over various lists of "the worst software bugs," I found that only a few of them directly involved these common pitfalls. The Mars Pathfinder bug was one of them. Initially flawless in its fulfillment of a very complicated mission, early on, the Pathfinder began experiencing periodic resets and the subsequent loss of data. The cause was a classic case of priority inversion.

To illustrate concurrency and how priority inversion can happen in an embedded system, I'll create a simple embedded system with one input, one output, a very fast logging device, and a graphical user interface (GUI). You can create such a system with or without concurrency. You could, for example, create software with a single loop as shown in Table 1. We have seen literally scores of systems designed like this. They are simple. They work. No concurrency is necessary in the software. But wait a minute. There is concurrency in this system. The user and the external device can change things at the same time. That means the system has concurrency. But does the software need concurrent threads to handle this? Well, that depends. If the input changes more quickly than our time to execute our simple loop, we have a problem. If it takes 100 ms to write to our graphics display and our external input device can do something every 20 ms, we would miss critical data while going through the loop.

So how do we solve that? We could start sprinkling I/O reads into the middle of the graphics library, but that is extremely messy. We could create two concurrent threads in our system: one to process the user interface (UI) and one to process the external device. In an embedded system, this could be accomplished in three ways: with a multitasking operating system (OS), with interrupts, or with hardware. In a multitasking OS, you would put a blocking delay at the end of the high-speed thread. In addition, you would enable the external thread to preempt the UI loop. This means the external device thread does not have to wait until the UI thread is complete before it can start. This could be done by giving the external device thread a higher priority than the user input thread. For example, if you are polling the external input every 10 ms, you would be "interrupting" the UI loop some place in the code every 10 ms. With interrupts on the external device, you would basically be doing the same thing without the need for a thread delay. You would be "event-driven" rather than periodically polling. But you would still be interrupting the UI thread every time the external device changed. Finally, you could solve this by using or creating hardware that could obtain the external data while the UI software is busy and go back to your single-thread solution. Table 2 provides a look at solving it using the multithreaded OS. Notice that the UI thread never stops but polls continuously (sometimes this is called the idle thread). The external device thread blocks (gives other threads the opportunity to run) and thus effectively polls the external input every 10 ms

| Our Example Design with Two Threads | |
|---|---|
| **User input thread** | **External device thread** |
| Check for user input | Check to see if something changed in the external world (input changed state or character received) |
| Process the user input | Process the changes from the input from external world and set the output accordingly |
| Update the display | Log the external data to our logging device when free |
| Log the user data to our logging device when free | Block for 10 ms |
| Rinse and repeat | Rinse and repeat |

Table 2—Steps that a user input thread and an external device thread may take in a system design with two threads

| Our Example Design with Three Threads | | |
|---|---|---|
| **User thread (low priority)** | **Network thread (medium priority)** | **External device thread (high priority)** |
| Check for user input | Check to see if there is data to be sent | Check to see if something changed in the external world (input changed state or character received) |
| Process the user input | Send the data over the Internet to the host | Process the changes from the input from external world and set the output accordingly |
| Update the display | Wait for the response for up to 100 ms | Log the external data to our logging device when free |
| Log the user data to our logging device when free | Block for 5 s+ | Block for 10 ms |
| Rinse and repeat | Rinse and repeat | Rinse and repeat |

Table 3—Steps that a user input thread, a network thread, and an external device thread may take in a system design with three threads

(plus our processing time).

We now have a complexity that both threads share the same logging device and cannot use it at the same time. So they would need to wait for the other to complete before use. But that is no problem since it is a fast device and thus neither thread will be unable to perform if it has to wait even the maximum amount of time (50 ms).

We deliver our masterpiece to marketing and they are happy, with one exception. They want to make this an Internet appliance. They want to send data periodically to a web server and to receive responses back from the web server. We will need to check to see if we have data every 5 s. Now, we have to add a third concurrent thread. This will be a medium-priority thread that sends the data to the Internet. It needs to respond more quickly than the UI but more slowly than the external data interface (see Table 3).

We have now set up the classic case for priority inversion to occur, and we will occasionally miss critical data (albeit rarely). Here is how it happens. The low-priority UI thread is writing to the log device when the high-priority device wants to write to it. So the high-priority thread blocks waiting for the device to become free. During that tiny window, the medium-priority thread preemptively interrupts the low-priority thread and runs for 100 ms thus blocking the high-priority thread from recording a critical transition of its input. Because it happens so seldom, this would probably never be found during testing. Only after we deliver the first 10,000 units to the field would this be reported.

## PREVENTING PRIORITY INVERSION

There are two primary ways of preventing this from happening. The first involves locking out preemption during critical sections of code. This can be done by locking out interrupts or through some OS-locking mechanism. In our example, we would place these locks in the low-priority thread around the use of the shared resource: the logger. When the low-priority thread logged our data, it would prevent anyone from preempting the low-priority thread until complete. This would prevent a higher-priority thread from preempting a lower-priority thread when it is dealing with a shared resource.

The second and less complex (for the designer) method is to let the OS prevent this from happening by using either priority inheritance or the priority ceiling techniques with shared resources. This would mean that any time a shared resource is used by any thread, that thread would take either the highest priority of the threads using the shared resource (priority inheritance) or the priority assigned to the resource (priority ceiling). There are problems with these features affecting latency, but they are beyond the scope of this article.

Any POSIX-compatible OS (including QNX, VxWorks, and Linux) has priority inheritance built in. It is well worth any real-time system designer's time to become very familiar with how this is used in their OS.

## MANAGING CONCURRENCY REQUIREMENTS

Most systems we design have concurrency requirements in them. It is our job as system designers to identify them and then create designs that robustly handle this concurrency—at times adding concurrency into our software. We need to be aware of the dangers of both implementing concurrency and the dangers that the solutions can cause as well. Next time, I will look at some more concurrency pitfalls in embedded systems.

*Bob Japenga has been designing embedded systems since 1973. In 1988, along with his best friend, he started MicroTools, which specializes in creating a variety of real-time embedded systems. With a combined embedded systems experience base of more than 200 years, they love to tackle impossible problems together. Bob has been awarded 11 patents in many areas of embedded systems and motion control. You can reach him at rjapenga@microtoolsinc.com.*

## RESOURCES

Computerworld, Inc., "Epic Failures: 11 Infamous Software Bugs," 2010, www.computerworld.com/s/article/9183580/Epic_failures_11_infamous_software_bugs?taxonomyId=18&pageNumber=1.

N. Dershowitz, The Blavatnik School of Computer Science, Tel Aviv University, "Software Horror Stories," www.cs.tau.ac.il/~nachumd/verify/horror.html.

S. Garfinkel, "History's Worst Software Bugs," *Wired*, 2005, www.wired.com/software/coolapps/news/2005/11/69355?currentPage=all.

M. Jones, Microsoft Corp., Microsoft Research, "What Really Happened on Mars?," 1997, http://research.microsoft.com/en-us/um/people/mbj/mars_pathfinder/mars_pathfinder.html.

by George Novacek (Canada)

# Diode ORing

Diode ORing is a commonly used method for power backup. But there's a lot more behind the method than meets the eye. This article describes some solutions for maintaining uninterrupted power.

In my article series, "Design a Robust Power Supply," (*Circuit Cellar* 258–259, 2012), I showed two blocking diodes, D1 and D2, forming an OR gate at the power input. Their purpose was to enable two primary power sources to be used to prevent the controller loss of power should one power source be interrupted. The principle of using diodes as an OR gate is neither new nor limited to systems with two primary power sources. You can see it whenever a power backup is needed, such as for volatile RAM, real-time clock (RTC), and so forth. George Martin's 2011 article "Design Development (Part 4): Processors, Power, and Interfacing" shows one example (*Circuit Cellar* 256).

## UNINTERRUPTED POWER

Let's consider some often overlooked aspects behind diode ORing for maintaining uninterrupted power. Whenever a device is powered from an external power distribution bus, a blocking diode is needed to isolate the internal storage capacitor from discharging into the bus. Two such diodes form an OR gate. Figure 1 shows the basic topology we frequently encounter.

In a safety-critical embedded control system, power sources A and B are usually batteries with their own generators and chargers. We can also see a similar arrangement in computers where power source A comes from the main power supply, for instance, while power source B is a battery to keep the RTC circuitry running.

Is the storage capacitor C1 sufficiently large enough to eliminate the need for the secondary power source? Yes, but not always. Assume, for example, that you have to ensure that an aircraft embedded controller survives a 1-s power interruption to maintain vehicle safety. This is a common requirement.

A typical aircraft system uses a 28-VDC power distribution system. Assuming there are no special operating voltage range requirements (such as those in my aforementioned *Circuit Cellar* article series) making the design even tougher, the lowest operating voltage to the controller will be 17.4 VDC, considering the blocking diode's (D1 or D2) forward voltage drop of about 0.6 V. A common switching regulator will operate reliably down to 16 V and the system may draw about 1 A.

For the system already operating at its lowest power input voltage of 17.4 V, the storage capacitor must provide sufficient capacity to hold the voltage for 1 s with a maximum drop

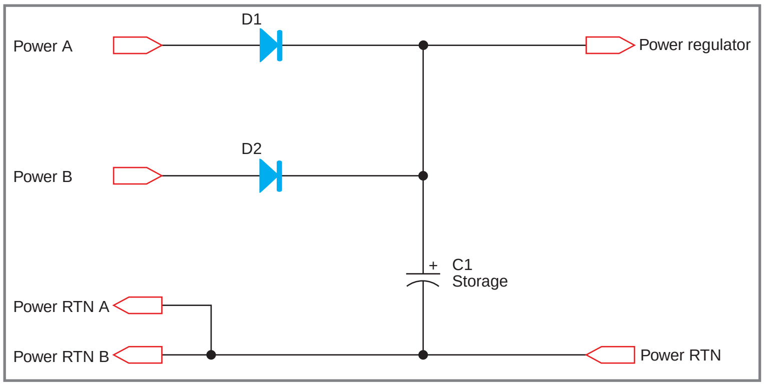



Figure 1—Power input with diode OR circuit

of 1.4 V.

The following equation expresses the relationship between the storage capacitor's voltage (V), capacitance (C), discharge current (i), and time (t):

$$V = \frac{1}{C}\int_0^t i \times dt$$

Considering the relatively small voltage drop of 1.4 V, the equation can be linearized to:

$$C = \frac{i \times t}{V}$$

This will obtain the capacitor value of C = 0.7 F. This is huge and hardly practical for many applications.

## SUPERCAPACITORS

Supercapacitors are currently available in a capacitance range up to 100 F, but their working voltage is typically only in the range of 2.5 to 2.7 V. To satisfy requirements for the 28-V system, plus headroom for inevitable spikes and derating, you would need to build an assembly of about 18 capacitors in series to obtain a 48-VDC working voltage. Depending on the value of the selected supercapacitor, several of those series capacitor assemblies would have to be connected in parallel to create the required 0.7 F/48 WVDC capacitor. The cost, size, and weight of the assembly are likely to be prohibitive. Worse, based on the present state of the art, the reliability of the assembly would likely be unacceptably low. Therefore, the solution with two power sources and two blocking ORing diodes is usually preferred. Strictly speaking, there would be no further need for the storage capacitor C1 due to the two rapidly switching ORing diodes. But, it is good engineering practice to provide large enough capacity to maintain operation for 30 to 50 ms. Such a capacitor would still be a hefty 33,000 µF/48 WVDC (33G), but this is manageable.

## THE NUMBERS GAME

Unfortunately, with diode ORing there is a problem lurking in the background. Those diodes represent dormant failures because, if used as shown in Figure 1, they cannot be tested during operation. A dormant failure means that it is present without the controller's diagnostics knowing about it and, therefore, an appropriate corrective action cannot be taken. When the second diode fails, it will bring the controller down. Diodes generally fail open or short with close to a 50/50 ratio. It depends on the whose data is consulted. Should one diode be shorted, its failure eliminates the blocking function. A power glitch on the distribution line with the faulty diode will bring the controller down, possibly also blowing the other diode by excessive current.

"Whenever a device is powered from an external power distribution bus, a blocking diode is needed to isolate the internal storage capacitor from discharging into the bus. Two such diodes form an OR gate."

Should the diode fail open, a power glitch on the other line will bring the controller down.

What can be done about it? The simplest and the most common method is sometimes called the "numbers game." Based on historical data, the diode's statistical reliability is calculated. The calculated results will vary depending on the operating parameters chosen, but it can be stated that a well-rated diode can exhibit better than $1.276 \times 10^9$ mean time between failures (MTBF). That amounts to one failure in 20,810 years of continuous usage. As there are two diodes in the controller and the controller is used on average, say, 4 hours every day, approximately one failure should be expected in 62,430 years. For all intents and purposes, this will probably never occur, but remember, there is no guarantee that failure will not occur in the first or the last second of operation during that period. More importantly, MTBF will quickly deteriorate with quantity. If you build 10,000 devices, which are not many by today's standards, you can expect one failure every 6.25 years among the devices' population.

In many industrial and safety-or-mission-critical products, where the manufactured quantity is often less than 1,000 units, the numbers game is acceptable. But what can you do when it is not?

You need to test the diodes. One approach is to call for regular maintenance checks, usually once a year, but scheduled maintenance is often frowned upon. It's a costly nuisance and, therefore, maintenance by demand is preferable.

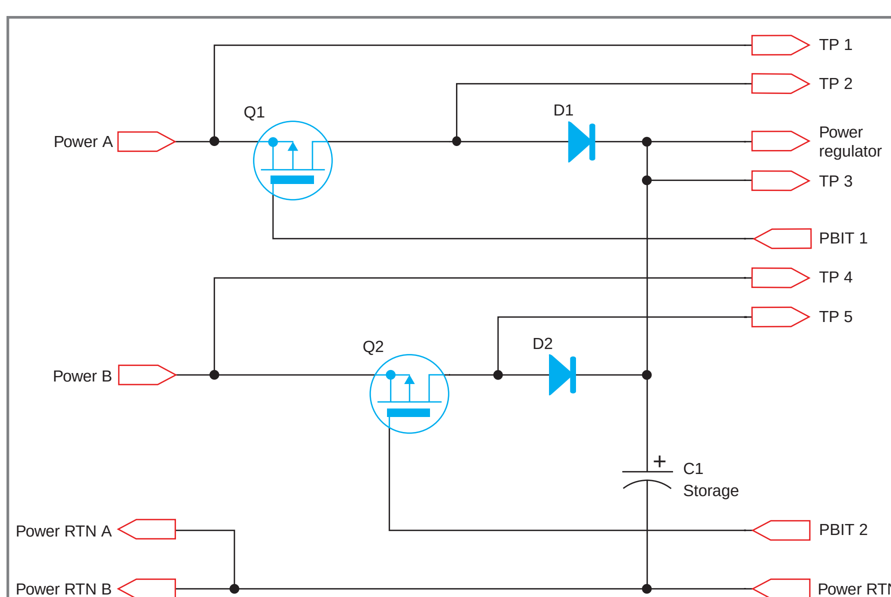



Figure 2 —Adding MOSFET switches Q1 and Q2 facilitates D1 and D2 integrity testing

That means maintenance is in response to a detected problem. The approach depicted in Figure 2 shows one solution. By adding MOSFET switches and test points for the controller's built-in test equipment (BITE), you can test the integrity of each diode. Such a test is hard to perform continuously in the background, so you may still want to play the numbers game and rely on the statistics while frequently performing the tests. Usually a test during the power-up built-in test (PBIT) is sufficient for the entire mission. Why? Because now you know that every mission is being started with both diodes intact. There is no dormant failure. A dual failure—that is, one diode and one power bus drop—would be needed during a mission to shut the system down. Because the probability of multiple failures diminishes exponentially, the risk to the system becomes astronomically small. Let's say, for example, the probability of either diode failing during the mission is $10^{-7}$ per 1,000 hours and the probability of the power supply hiccup is $10^{-4}$ per 1,000 hours. The probability of both a diode failure and the supply outage between two tests would then be $10^{-11}$, where $10^{-9}$ is generally accepted as safe. (This is a simplified example for illustration only. In reality, you would have to consider the failures of the MOSFETs, the BITE, and so forth.)

## SEEMINGLY SIMPLE SOLUTIONS

Many design challenges can be solved by seemingly simple, inexpensive solutions. But beware of those simple solutions. Their simplicity can be deceptive. Here, the 1-s power outage challenge was solved without the need for a huge capacitor assembly. At the same time, however, a can of worms was opened by introducing dormant failures. This once again demonstrates that there is a lot more going into electronics design than meets the eye. Ensuring a product's reliability often takes more effort than the design of its functionality and features.

*George Novacek (gnovacek@nexicom.net) is a professional engineer with a degree in Cybernetics and Closed-Loop Control. Now retired, he was most recently president of a multinational manufacturer for embedded control systems for aerospace applications. George wrote 26 feature articles for* Circuit Cellar *between 1999 and 2004.*

## RESOURCES

G. Martin, "Design Development (Part 4): Processors, Power, and Interfacing," *Circuit Cellar* 256, 2011.

G. Novacek, "Design a Robust Power Supply (Part 1): Power Dissipation," *Circuit Cellar* 258, 2012.

———, "Design a Robust Power Supply (Part 2): Abnormal Surge Voltage," *Circuit Cellar* 259, 2012.

by Ed Nisley (USA)

# MOSFET Channel Resistance

## Theory and Practice

This article describes the basics of power MOSFET operation and explores the challenges of using a MOSFET's drain-to-source resistance as a current-sensing resistor. It includes a review of fundamental enhancement-mode MOSFET equations, compares them with Spice simulations, and shows measurements from an actual MOSFET.

In this day and age, linear voltage and current regulators have given way to switched-mode regulators combining digital logic with power MOSFETs. Because the transistors operate either fully on or fully off, they dissipate much less power than linear-mode regulators.

However, switched-mode regulators must measure and control the current through those transistors in order to produce the correct output. The most common regulator configuration passes the current through an additional low-value resistor that produces a voltage proportional to the current. Some topologies put one resistor terminal at the circuit ground potential, while others use a high-side resistor with a differential amplifier. Either configuration works well, as demonstrated by the vast number of switching regulator ICs using them, but those sense resistors can waste a significant amount of power: $P = I^2R$ doesn't take a vacation just because you only need $V = IR$!

I've been pondering a low-voltage, high-current, battery-powered LED light, where that wasted power would reduce the light's run time. It occurred to me that MOSFET power transistors act as reasonably linear resistors when they're fully turned on, so perhaps I could eliminate the current-sense resistor and measure the voltage drop across the MOSFET's inherent resistance.

This is not a new idea, by any means, and several power-supply ICs implement resistorless current sensing. Because my application didn't suit those ICs, however, I decided to measure the actual behavior of some power MOSFETs and compare those numbers with datasheet specifications to understand what was going on.

I'll start by reviewing some fundamental enhancement-mode MOSFET equations, compare them with Spice simulations, and then show measurements from an actual MOSFET. In upcoming columns I'll describe an Arduino-based "curve tracer" that can measure a MOSFET's resistance $R_{DS}$ for reasonable values of both drain current $I_D$ and gate voltage $V_{GS}$. The tester controls the transistor temperature using a Peltier module that I'll describe along with a summary of MOSFET measurements.

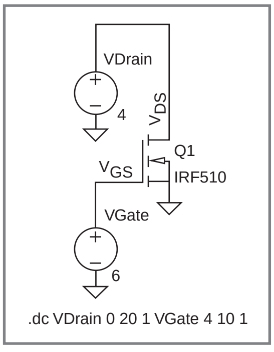



Figure 1—A simple Spice model applies voltage sources with unlimited current capacity to the gate and drain terminals: don't try this on your workbench!

### POWER MOSFET BASICS

The extremely simple Spice model in Figure 1 applies ideal voltage sources to an IRF510

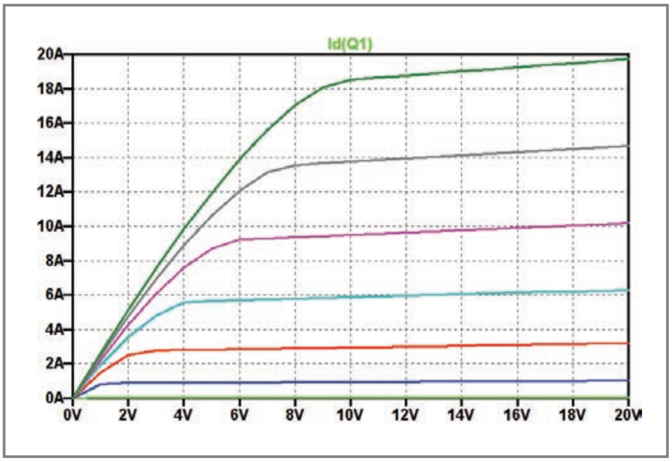



Photo 1—Modeling a classic IRF510 MOSFET produces nice, clean drain current curves that far exceed the transistor's 5.6-A current rating as the gate voltage steps from 4 V to 10 V.

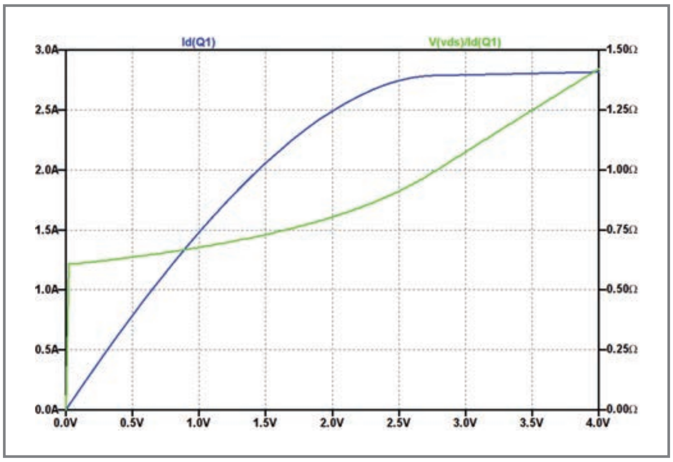



Photo 2—The blue trace shows $i_D$ varying with $v_{DS}$ for $V_{GS}$ = 6 V. The green trace plots the ratio of $v_{DS}/i_D$, which gives the MOSFET's effective resistance $r_{DS}$.

MOSFET model's Gate and Drain terminals, with the Source terminal at circuit common. Sweeping both voltage sources generates the classic curve tracer plot of drain current $I_D$ against drain voltage $V_{DS}$ in Photo 1, with $V_{GS}$ increments producing the neatly stepped lines.

Don't try this on your lab workbench, however, because the 400-W power dissipation in the upper right corner of that plot exceeds the IRF510's spec by an order of magnitude. The 20-A drain current at that point also exceeds the MOSFET's 5.6-A limit, so be careful when interpreting simulation results! I chose an IRF510 for this model because it has a relatively high 540-mΩ resistance that's easy to see in the simulations, but, as we'll see later, a more modern MOSFET with a much lower drain resistance will minimize the power dissipation.

The curves in Photo 1 have three distinct regions, each corresponding to a different MOSFET operating condition: active, cutoff, and linear. Although even a reasonably accurate Spice model represents an idealized version of reality, understanding what's going on inside the MOSFET requires a trip into even more idealized equations.

In what follows, lowercase letters represent instantaneous values and measurements; uppercase letters represent constants, ranges, and graph axes; and capitalized subscripts represent transistor terminals or conditions. For example, $v_{GS}$ is a measured gate-to-source voltage, $V_{GS}$ is a preset value, and $V_T$ is a MOSFET's specified threshold voltage.

The conspicuous lines laddering up the right half of Photo 1 represent the MOSFET's *active* region, where the drain current varies almost exactly as the square of the gate voltage, as predicted by this equation:

$$i_D = K\left(v_{GS} - V_T\right)^2$$

That equation applies only for $v_{DS} > (v_{GS} - V_T)$, which means that the leftmost ends of the relatively horizontal lines form a parabola curving upward from the origin. The simple equation does not include terms accounting for the upward tilt arising from increasing $v_{DS}$; in fact, $v_{DS}$ doesn't appear in the equation at all.

The constant K (also known as $K_P$, the *process transconductance*) collects a handful of MOSFET process parameters, ranging from electron mobility to channel length, into a single number with a unit of siemens = 1/Ω. Because datasheets don't include any of those numbers, there's no way to calculate K for any given MOSFET, although you can get a rough approximation from actual measurements or the Spice model. You can cheat by pulling the numbers directly from the model's source code, but that won't help in real life.

The threshold voltage $V_T$ represents the minimum gate voltage required to form a conductive channel between the MOSFET drain and source terminals, so the *cutoff* region consists of the single line along the $I_D$ = 0 A axis where $v_{GS} < V_T$. In this region, the MOSFET channel (or, more precisely, the volume of silicon where the channel would be with higher $v_{GS}$) conducts very little current, typically well below 100 µA, and the device dissipates very little power. The bright green line at ID = 0 in Photo 1 corresponds to $V_{GS}$ = 4 V, at the upper edge of the IRF510's $V_T$ = 2 V to 4 V specification.

Finally, the graceful arcs connecting the origin to the active region in Photo 1 represent the MOSFET's *linear* region, where $v_{DS} < (v_{GS} - V_T)$ and the drain current follows this equation:

$$i_D = K\left[2\left(v_{GS} - V_T\right)v_{DS} - v_{DS}^2\right]$$

The first term shows that the current varies linearly with the drain voltage, with the second term becoming increasingly important as $v_{DS}$ increases: the linear region is only approximately linear. For a given $v_{GS}$ in this idealized world, the MOSFET behavior transitions neatly between its linear and active regions when $v_{DS} = (v_{GS} - V_T)$. In all regions, the MOSFET power dissipation is:

$$p = v_{DS}\, i_D$$

As I mentioned earlier, that value can be surprisingly high in the active region. Venturing outside the transistor's

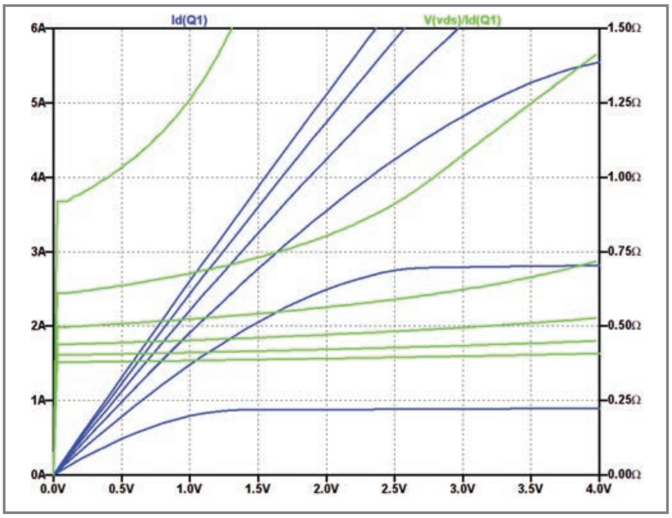



Photo 3—The linearity of the green curves plotting drain resistance $R_{DS}$ improves as $V_{GS}$ steps from 5 V to 10 V. Blue drain current $I_D$ has $V_{GS}$ = 5 V on the bottom curve, green resistance $R_{DS}$ has $V_{GS}$ = 5 V on the top.

safe operating area defined in the datasheet will eventually release magic smoke from the package, although the power limit varies depending on duty cycle and repetition rate.

Homework: Plot constant power contours on Photo 1 at 100 mW, 1 W, 10 W, and 100 W.

Linear voltage and current regulators operate in the active region, as do fancy Class A audio amplifiers. The latter generally run barely inside the transistor's maximum allowable DC power dissipation contour, imposing small AC signals on large DC biases, with huge heatsinks controlling the MOSFET junction temperature: cooling fans mix poorly with high-end audio gear.

Switching regulators use MOSFETS as on-off switches that either pass current from the bulk supply or cut it off entirely. The On state operates in the MOSFET linear region, where other circuit components limit $i_D$ and the transistor dissipates relatively little power. The Off state corresponds to the MOSFET cutoff region, where the very small leakage current results in low power dissipation. Much of the difficulty surrounding switching power supply designs involves reducing the amount of time spent traversing the active region while switching between on and off!

## SEEKING LINEARITY

The blue trace in Photo 2 zooms in on the simulated drain current at constant $V_{GS}$ = 6 V with drain voltages up to 4 V. The green curve plots $i_D/v_{DS}$, the MOSFET's effective drain-to-source resistance $r_{DS}$, which shows the expected linear region near the origin and the neatly faired junctions to the active region.

The resistance will be most constant when the $v_{DS}^2$ term is small compared to the first term: $2(v_{GS} - V_T)$ must be much larger than $v_{DS}$. The IRF510 has a threshold voltage $V_T \simeq 3$ V, so $2(v_{GS} - V_T) \simeq 6$ V in Photo 2. You can see the $r_{DS}$ curve beginning to tilt upward between $v_{DS}$ = 1.0 V and 1.5 V, so, at least in this case, "much larger" means more than a factor of four.

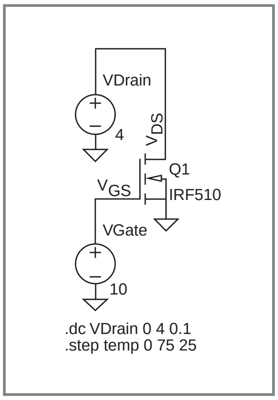



Figure 2—The ".step" Spice command varies the simulated temperature for each simulation run.

Photo 3 shows the behavior of $i_D$ (blue) and $r_{DS}$ (green) with $V_{GS}$ stepping from 5 V to 10 V: higher $V_{GS}$ produces the upper $i_D$ curves and the lower $r_{DS}$ curves. As expected, the drain resistance varies dramatically when $V_{GS} \le 7$, but the MOSFET becomes a nicely linear resistor with $V_{GS} \ge 8$ V: $r_{DS} \approx 400$ mΩ for $v_{DS}$ below 3 V.

As with all MOSFETs operating in the active region, the external circuit must limit the drain current. Without that limit, a MOSFET with $r_{DS}$ = 400 mΩ will draw 5 A from a 2-V supply, which is probably not what you want. The inductor in a typical switching power supply delivers a more or less linearly increasing current from the bulk voltage source, with a comparator turning the transistor off when the current reaches a predetermined maximum value that's much lower than the DC limit predicted by $r_{DS}$.

The LED lighting application I'm considering would use the MOSFET as a resistor to both control and measure the current from a battery through an LED string, rather than use it as a digital switch. As a result, the firmware must set $v_{GS}$, measure the resulting $v_{DS}$ voltage, then feed those values into a MOSFET model to find the corresponding $i_D$. Simply setting $V_{GS}$ = 10 V, however, would probably vent magic smoke from a string of four amber LEDs connected to a 7.4-V lithium battery: knowing that $i_D$ reached 3.5 A just before the LEDs vaporized would not be particularly helpful.

Unfortunately, all transistor (and LED!) parameters

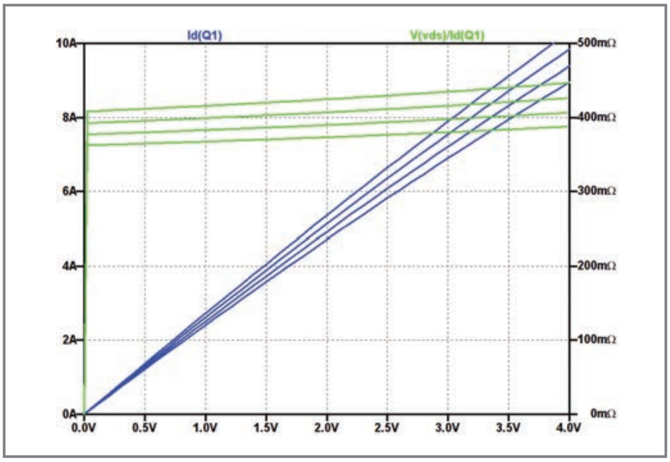



Photo 4—All MOSFET parameters depend strongly on temperature, as shown by these current and resistance variations from 0°C to 75°C, all with $V_{GS}$ = 10 V.

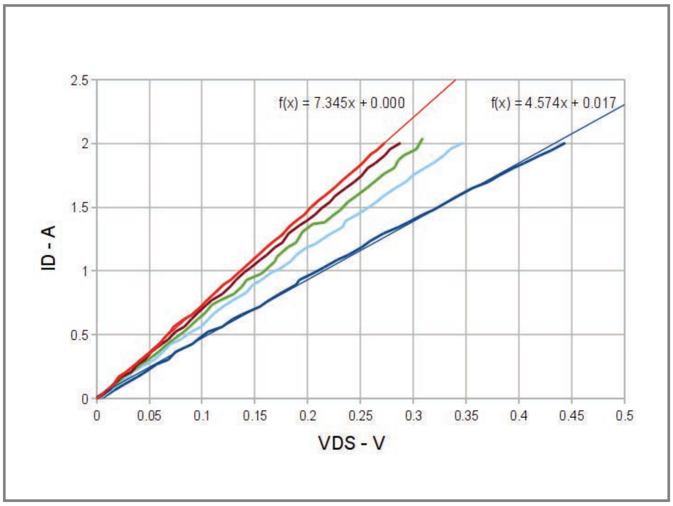



**Photo 5**—Measurements of $R_{DS}$(on) for a BUZ71A transistor show good agreement with its datasheet value. The slope of the linear-fit lines gives the conductance $G = 1/R_{DS}$. With $V_{GS}$ = 10 V, $R_{DS}$ = 140 mΩ instead of the rated 120 mΩ, but the measurement includes both connector and wiring resistance.

depend strongly on temperature, so a simple equation doesn't capture all the details. The Spice model in Figure 2 holds $V_{GS}$ constant while varying both $v_{DS}$ and the simulated temperature, so each quartet of traces in Photo 4 represents the variation in $i_D$ and $r_{DS}$ with temperature, all for $V_{GS}$ = 10 V. LTSpice can vary up to three different parameters at once, but I'll leave the resulting mesh of traces to your imagination.

The drain current at $v_{DS}$ = 3 V ranges from 8 A at 0°C to 7 A at 75°C. Because MOSFET $r_{DS}$ increases with increasing temperature, the drain current decreases for constant $V_{DS}$. That makes MOSFETs less likely to experience thermal runaway than junction transistors; you can still destroy a MOSFET, but it's more difficult.

Fortunately, at least over the relatively small temperature range I'm interested in, $r_{DS}$ varies almost linearly with temperature, at least over a total range that doesn't require precise measurement. A simple thermistor circuit, rather than a complex thermocouple, ought to produce useful results.

## REAL NUMBERS

Although simulations can help eliminate bad designs without blowing up parts, it's always useful to confirm those results with actual measurements. I don't have an IRF510 in my collection, so I used a BUZ71A logic-level MOSFET with a nominal $V_T \leq 4$ V and maximum $R_{DS}$ = 120 mΩ. The test circuit limits the drain current with another MOSFET (in power-wasting active mode!) and measures the actual current by sampling the voltage across a (power-wasting!) sense resistor. I used a 100-mΩ sense resistor to generate this data, but I'll probably use a larger value to improve the signal-to-noise ratio in the final circuit.

Stepping $v_{GS}$ from 6 V to 10 V covered the MOSFET's linear region of operation, as shown by the traces in Photo 5. Because the BUZ71A has a lower $R_{DS}$ than the simulated IRF510, the drain voltage will be smaller at a given current. The 2-A upper limit for $i_{DS}$ in that graph indicates the limitations of my power supply and MOSFET heatsink.

The reciprocal of the linear-fit line slope gives $R_{DS}$, which varies from 220 mΩ with $V_{GS}$ = 6 V to 140 mΩ with $V_{GS}$ = 10 V and is in good agreement with the specified 120-mΩ maximum value. Unlike Spice simulations, the actual voltage measurement includes the IR drop along the MOSFET connections: the resistance calculated from the voltage and current will include connector and wiring losses.

Using an ordinary MOSFET as a current-sense resistor poses some challenges, but the early results look promising.

## CONTACT RELEASE

You can buy four-terminal MOSFETs with an internal current mirror that presents a fixed fraction of the drain current to the additional sense terminal, but these tend to be specialized for particular applications. If you're designing one of those applications, of course, then a current-mirror MOSFET could be the right tool for the job.

For the rest of us, an ordinary MOSFET may serve just as well. In my next column, I'll describe an Arduino-based curve tracer that measures $R_{DS}$ in the linear region for relatively low-power applications. ■

*Ed Nisley is an EE and author in Poughkeepsie, NY. Contact him at ed.nisley@ieee.org with "Circuit Cellar" in the subject to avoid spam filters.*

## PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/263.

## RESOURCES

V. Barkhordarian, International Rectifier, "Power MOSFET Basics," www.irf.com/technical-info/appnotes/mosfet.pdf.

International Rectifier, "Application Note 957: Measuring HEXFET MOSFET Characteristics," www.irf.com/technical-info/appnotes/an-957.pdf.

Maxim Integrated Products, "MAX1858 Dual 180° Out-of-Phase PWM Step-Down Controller with Power Sequencing and POR," www.maxim-ic.com/datasheet/index.mvp/id/3533/t/al

ON Semiconductor, "NTMFS4833NS: SENSEFET 30-V 156 A 2.2 mΩ Single N-Channel SO-8FL," www.onsemi.com/PowerSolutions/product.do?id=NTMFS4833NS.

by Robert Lacoste (France)

# Radio Frequency Mixers

Frequency mixers are essential to radio frequency (RF) designs. They are responsible for translating a signal up or down in frequency. This article covers the basics of RF mixers and their real-life applications.

Welcome back to the Darker Side. This month I'll present a topic that frightened me for years: mixers—more precisely, radio frequency (RF) mixers. As a kid, when I started playing with a soldering iron, building alarms or other gadgets, my father gave me hundreds of old radio magazines. They were full of short-wave receiver schematics, with strange names like "intermediate frequency" (IF) and "super-heterodyne." I tried to understand, but for a long time I safely restrained myself to the digital side. In fact, since using 7400 and 4011 logic chips was comparatively easy, moving to slightly more complex systems built around 6502 or Z80 chips was not so hard. I started playing with RF only 15 years later. Discovering how a mixer actually works was a revelation!

## WHAT IS A MIXER?

RF usually means very high frequencies. I know there are very low frequency (VLF) systems in service to communicate with submarines, but let's stay generic. The carrier used to transmit the information over the air has a frequency orders of magnitude higher than the information stream itself. For example, FM is broadcast around 100 MHz, whereas the stereo audio channel uses only some tens of kilohertz of bandwidth. Therefore, it is useful to translate a slice of the frequency spectrum from low frequencies to high frequencies for a transmitter, or from high frequencies to lower ones for a receiver. That's where frequency mixers shine. A frequency mixer is a frequency translation device that could be used either to move up (up convert) or down (down convert) any part of the spectrum.

Let's start by considering frequency mixers as black boxes. A mixer has one signal input port, one so-called local oscillator (LO) input port, and one signal output port, as shown in Figure 1. For a down converter, the input is traditionally named RF and the output is IF. This is the opposite for an up converter.

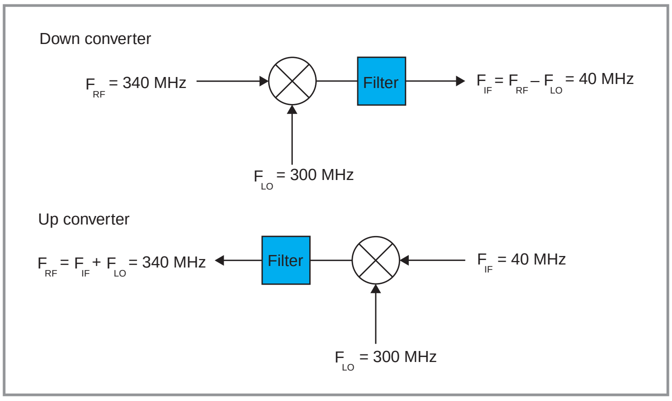



Figure 1—A mixer is a three-port device: a signal input, an LO input, and a signal output. When used with a frequency filter, it generates either a sum or a difference frequency. It can then increase (up convert) or decrease (down convert) the frequency of an RF signal.

What is the behavior of an ideal frequency mixer? Given an input frequency $F_{IN}$ and an LO frequency $F_{LO}$, it generates a sum of two signals on its output, with respective frequencies of the sum $F_{IN} + F_{LO}$ and the difference $F_{IN} – F_{LO}$ (or the opposite, whichever is positive) of the original frequencies. You can then filter the output signal to recover the component you are looking for. For example, a low-pass filter with select $F_{IN} – F_{LO}$.

You must know that a mixer is fundamentally a linear device, as long as it is used correctly. Its output will be doubled in amplitude if you double the amplitude of the input signal, and the phase relationship between input and output is maintained. Moreover, mixers—at least passive ones, as you will see—are reversible devices. With the same mixer you can input a signal on its IF port and use the RF port as an output, or the opposite.

You must understand that a mixer is one of the key building blocks of nearly all RF systems, used as a frequency translator or as a modulator (with the IF signal modulating the LO one). If you want to know how they work, just keep reading.

## NONLINEARITY HELPS

A trigonometrical identity is fundamentally the basis of all frequency mixers. Take a deep breath and repeat after me: 2.cos(a).cos(b) = cos (a+b) + cos (a-b).

I'm sure you learned this one, no? If you prefer a graphical version, see Figure 2, which shows a 300-MHz sine wave multiplied by a 340-MHz sine wave. The result is a slow 40-MHz sine with an added high-frequency, 640-MHz sine. That's it. Basically multiplying two sine waves generates a signal that is the sum of two sines, respectively, at sum and difference frequencies. Great. That's exactly what we were looking for!

Therefore, you can build a frequency mixer with any electronic part that "multiplies" two input voltages, and they are numerous. More precisely, it will work as long as you use any nonlinear component. Why? Imagine that you have two input voltages, $V_1$ and $V_2$. Add them and send them through a nonlinear device. Their output will be a nonlinear function of $x = V_1 + V_2$, say $f(V_1 + V_2)$. But you can almost always approximate this function by a second order equation $f(x) = a + bx + cx^2$. If you do the math, you will see that the last term will inevitably include the product $V_1 \times V_2$ of the two input voltages. So any nonlinear device could be used as a mixer, but of course, some are more effective or "cleaner" than others!

## MIXER INTERNALS

What is the simplest nonlinear electronic device? It is a diode, as the current going through it is not proportional to the voltage applied on its terminals. That brings us to the old so-called unbalanced diode mixer. Its schematic, as well as a simulation I made using Labcenter Electronics's Proteus CAD tool suite, is shown in Figure 3. To better understand this schematic, you must know that care is usually taken to set the LO signal power at a far higher power than the mixer input signal. The LO mainly defines the diode behavior. When the LO signal is positive enough, the diode is conducting. When the LO voltage is negative, the diode is in a blocking state. As the small input signal is added to the LO voltage, it is chopped by these alternating LO cycles: the input signal passes through when the LO is in its positive half period.

So, this means the input signal is multiplied by a square signal at the LO frequency. There is a kind of voltage multiplier, and this makes a mixer. However, as a square signal is more complex than a sine, it includes a fundamental but also harmonics. Therefore, such a diode-based mixer generates the required difference and sum frequencies as described, but also plenty of harmonics of both LO and RF signals as well as more complex combinations. In fact, you will find in the output any frequency that could be generated as $(n \times F_{LO} + m \times F_{RF})$, where n and

$$\cos(a) \times \cos(b) = \frac{1}{2}\left[\cos(A+B) + \cos(a-b)\right]$$

$$\cos(\Omega_{RF} \times t) \times \cos(\Omega_{LO} \times t) = \frac{1}{2}\left[\cos\{(\Omega_{RF} + \Omega_{LO}) \times t\} + \cos\{(\Omega_{RF} - \Omega_{LO})\} \times t\right]$$

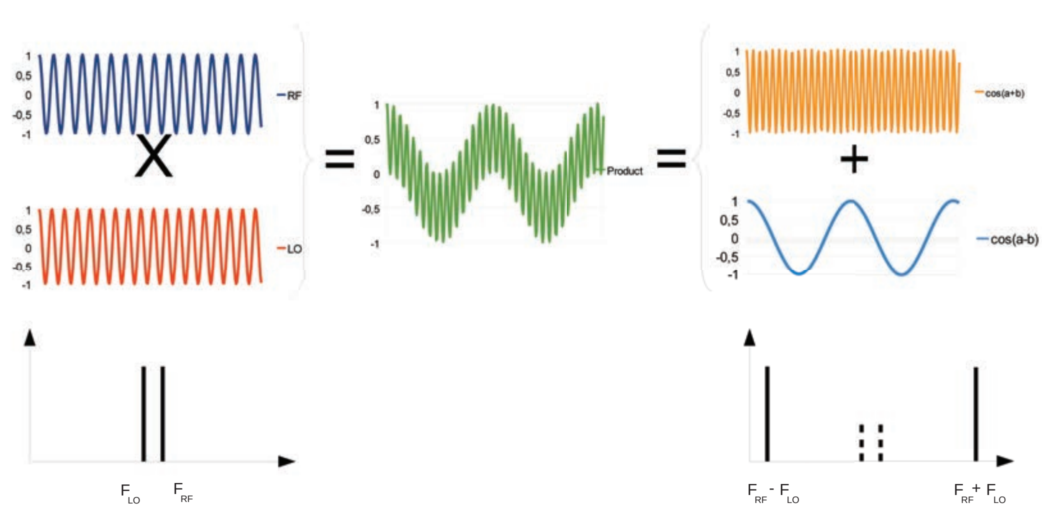



Figure 2—When two sine signals are multiplied, the result is the sum of two sines of sum and difference frequencies. This is the basic mechanism used by a frequency mixer. The OpenOffice spreadsheet used to generate this figure is available on *Circuit Cellar*'s FTP site if you want to play with it.

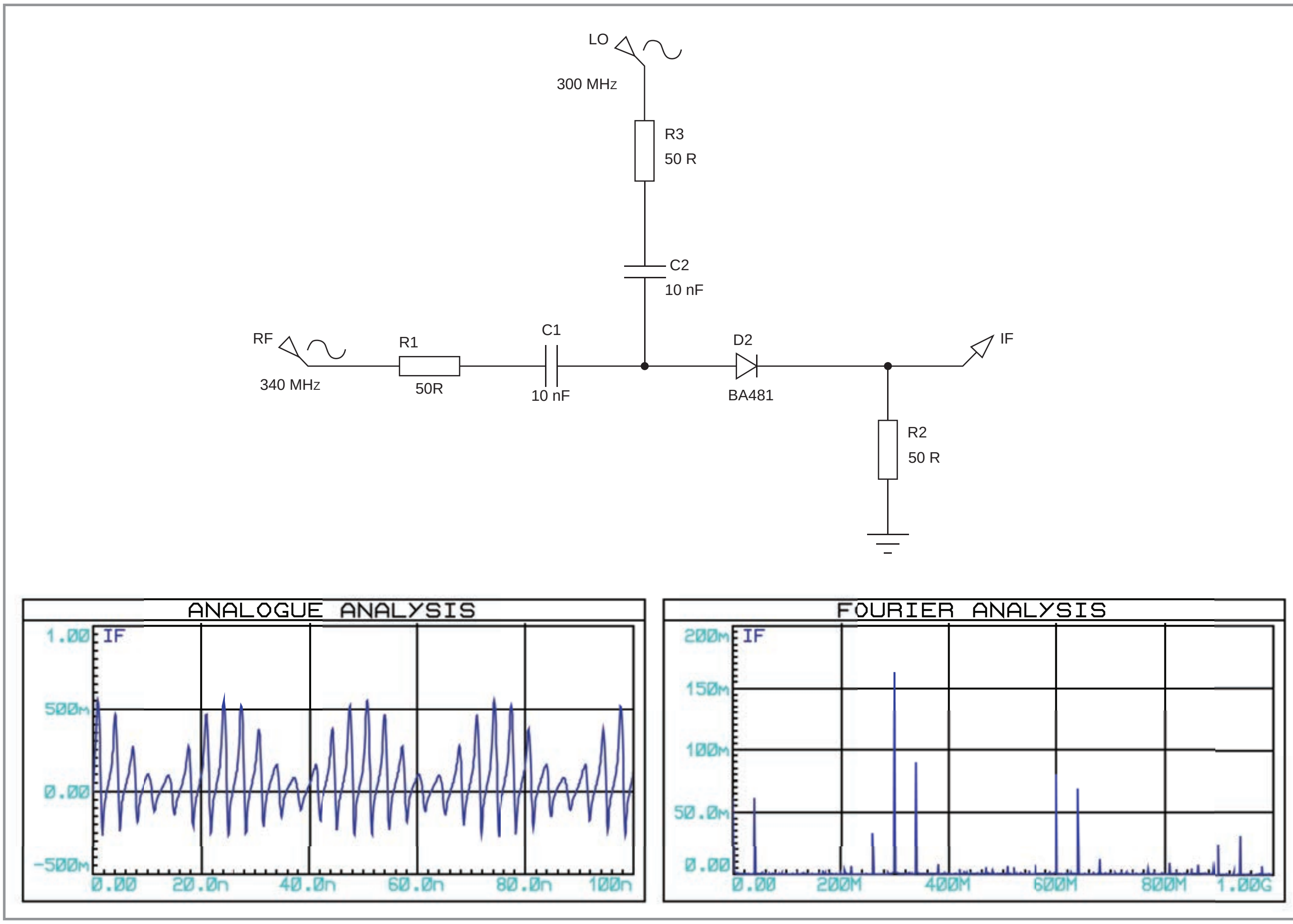



Figure 3—This is the schematic of a simple single-diode unbalanced mixer and its simulation done with Labcenter's Proteus. The RF and LO frequencies are 340 MHz and 300 MHz, respectively. You can see on the output spectrum that these two frequencies are still visible, as well as the difference 40 MHz and sum 640 MHz, among others.

m are arbitrary integer numbers, positive or negative. As anticipated, you will have frequencies $F_{LO} + F_{RF}$ and $F_{RF} - F_{LO}$. You'll also have stranger frequencies such as $2 \times F_{RF} - F_{RF}$ or even $7 \times F_{RF} - 5 \times F_{LO}$! As you will see, this makes mixers very interesting headache generators!

Even if they are used from time to time as illustrated in Photo 1, such single-diode unbalanced mixers have disadvantages. In particular, they enable some of the input signals (RF and LO) to pass through to the output. RF designers say their isolation is low. Therefore, more complex topologies were invented, such as the single-balanced mixer (which uses two diodes and strongly attenuates either the LO or RF signal but not both), or the very common double-balanced mixer. This one is built using four diodes and two hybrid transformers. It strongly attenuates both the RF and LO signals. Figure 4 is the schematic and simulation of a double-balanced mixer using the same RF and LO signals as shown on Figure 3. You will see that the output spectrum is far cleaner, with the difference and sum frequencies well visible even if there are still plenty of other harmonics but at lower levels. Take care. This simulation assumed perfect transformers. Real life is a little more complex, as you will see shortly.

How does such a double-balanced mixer work? Just as for the unbalanced mixer, the LO signal, which has a strong power, successively turns on each arm of diode pairs. In turn, this swaps the polarity of the connection between the input and output transformers, which provides the same kind of chopping as with the single-diode configuration but

Photo 1—This is an example of a single-diode mixer, extracted—if I remember correctly—from a 1965 HP8551 spectrum analyzer. The diode is the small cartridge. It is encapsulated inside the BNC connector, which is visible on the back.

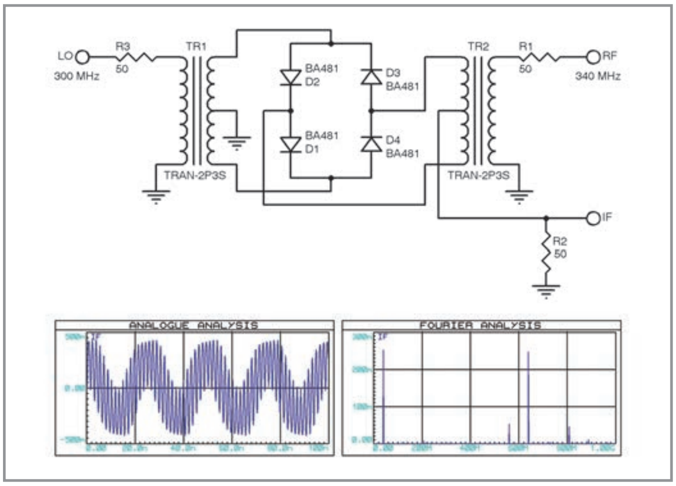



Figure 4—A double-balanced mixer is a little more complex, but its output spectrum is far cleaner. Compare it with Figure 2 to be convinced!


Photo 2—For my experiments, I used this M/A-COM Technology Solutions RF mixer. It is specified up to 4 GHz.

with a far better isolation and reduced spuriousness, thanks to the balanced architecture.

## MIXERS IN REAL LIFE

Simulations are great, but let's see what a mixer looks like in real life. For this experiment, I used a wideband M/A-COM Technology Solutions MD-525-4 double-balanced mixer I had on my shelf (see Photo 2). I hooked its LO port to a lab-grade RF generator set to a frequency of 300 MHz and a power of 10 dBm:

$$10^{\frac{10}{10}} = 10 \text{ mW, or an RMS voltage of}$$

$$U = \sqrt{(P \times R)} = \sqrt{(10 \text{ mW} \times 50\ \Omega)} = 0.7 \text{ V}$$

I then connected the RF input port to another lab generator set at a frequency of 340 MHz and with a varying signal power. Lastly, I connected my HP71200C (now Agilent Technologies) spectrum analyzer to the IF output port and looked at the result. Photo 3 shows the mixer output when the RF input signal power is set to –20 dBm (10 µW), –10 dBm (100 µW), or 0 dBm (1 mW). You can clearly see that the output spectrum is quite clean when the input power is very low, but it is more and more complex when it is increased. Why? Simply because the harmonics generated by a nonlinear device such as a diode are nonlinear. The amplitudes of the high-order spurious signals generated by the mixer climb more quickly than the amplitude of the input signal.

A good exercise is to try to figure out where each frequency of the mixer output comes from. In my example, the result is shown in Figure 5. Look at it carefully. This will help you understand what's going on.

Now you know that life is a little more difficult than you expected when you started to read this article. Unfortunately, all mixers are made with components such as diodes or transistors, so they all exhibit the same behavior. The output will never be as clean as you expected. Therefore, a mixer is usually used in conjunction with frequency filters to isolate the frequency you are looking for. The key is to be clever enough to select the RF, LO, and IF in such a way that this required frequency is not too close to high spurious signals. This is easy for fixed-frequency systems, but really complex when the working frequency is adjustable (e.g., on a wideband radio receiver). There are calculation tools that will help you with that task, such Hittite Microwave's free mixer spur chart calculator.

## SELECTING A MIXER

How should you select a mixer for your next RF project? You could, of course, dig through the catalogs of mixer specialists (i.e., Analog Devices, Hittite Microwave, Maxim Integrated Products, Mini-Circuits, Synergy Microwave, etc.), but what is important?

The first key parameter is, of course, the frequency range—more exactly, the frequency ranges—as each mixer is specified for a given range of frequency on each of the ports: RF, LO, and IF. This starts at sub-megahertz values and can climb very high in the microwave region, up to some tens of gigahertz for surface-mount technology (SMT) devices. Don't forget that RF and IF ports can be swapped at least for passive mixers. (Some mixers are "active" in the sense that they include an amplifier on the RF and/or LO path.) You will also have to check the characteristic impedance of the mixer ports. Most mixers are designed for standard 50-Ω impedance on all ports, but this is not always the case, in particular, for integrated circuits. In that case, you will have to design an impedance-matching circuit for the best performance.

The next important parameter is the LO level. Each mixer is specified for a given power that must be applied on the LO port. The rule is this: in order for a mixer to work properly, the LO power must be orders of magnitude higher than the highest possible input power. At least a 10-dB difference is a strict minimum if you accept a not-too-clean output spectrum. Therefore, you must check what the input signal power is in the worst-case condition and add 10 dB—or better 20 dB—to set the required LO power. You will soon discover that the best mixers require very high LO power, which makes power consumption a problem in a lot of designs, as you will need to generate very strong LO signals to drive them.

Next, check the mixer conversion loss. This is a measure of the frequency translation efficiency from RF to IF or vice versa. It is at least 3 dB, as the input power will be split between the sum and difference frequencies, but could be far higher. For standard passive mixers, you can expect losses around 6 dB to 9 dB, which means you will lose 80% of the signal power in the mixer. The losses are closely linked to the actual LO power. A mixer has its specified losses when you use it with its specified LO signal power. Nothing prevents you from using it with a slightly lower LO power, but then the conversion losses will be higher. Of course, these conversion losses can be negative (i.e., a conversion gain) for active mixers that have built-in amplifiers.

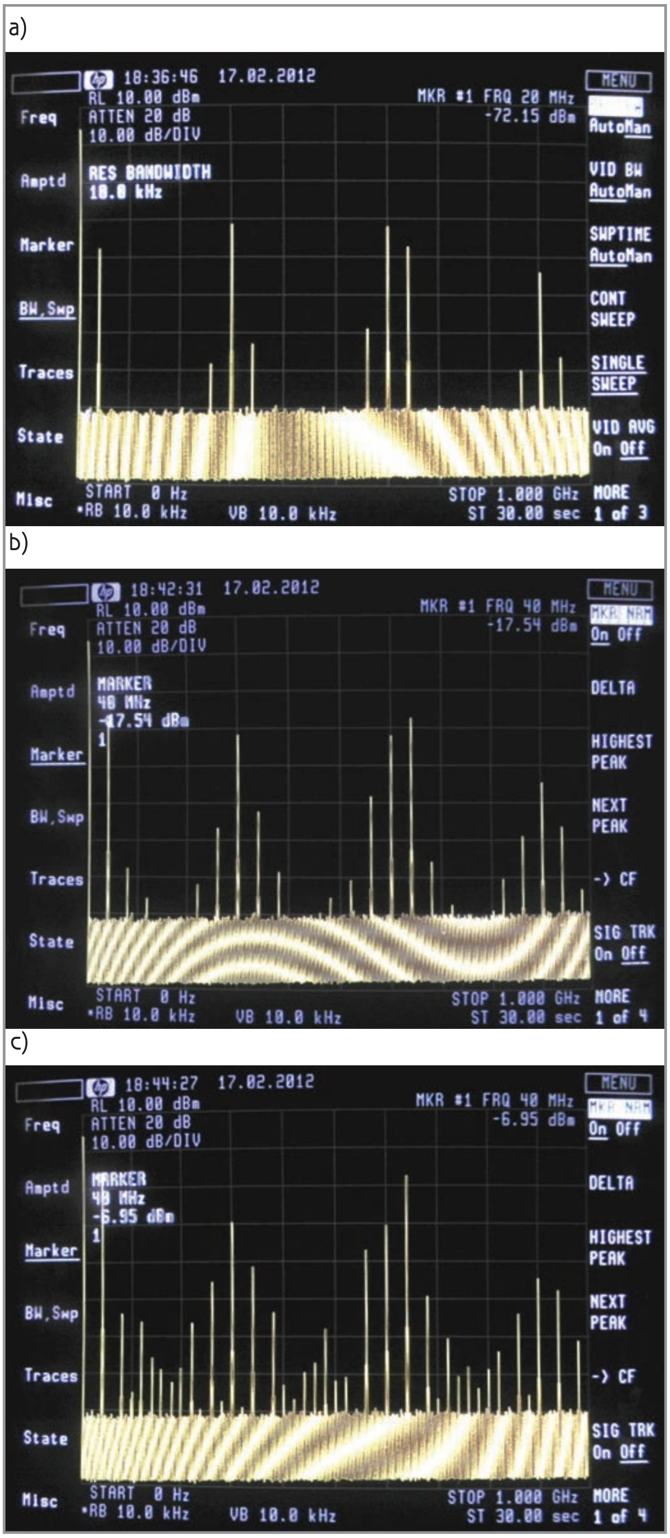



Photo 3a—This image shows the output spectrum from the M/A-COM Technology Solutions mixer, with a 340-MHz RF input that is set to –20 dBm and a 300-MHz LO. The horizontal scale ranges from 0 to 1,000 MHz, 100 MHz per division. The difference 40 MHz and sum 640 MHz are clearly visible, but the LO leakage (300 MHz and its harmonics) is high. b—This is the same experiment with a –10-dBm input signal. The spurious frequencies are higher and more numerous. c—Lastly, this is the output spectrum shown with 0-dBm input power. Plenty of nasty harmonics and spurious are visible in this image. It becomes difficult to isolate the required signal with a filter.

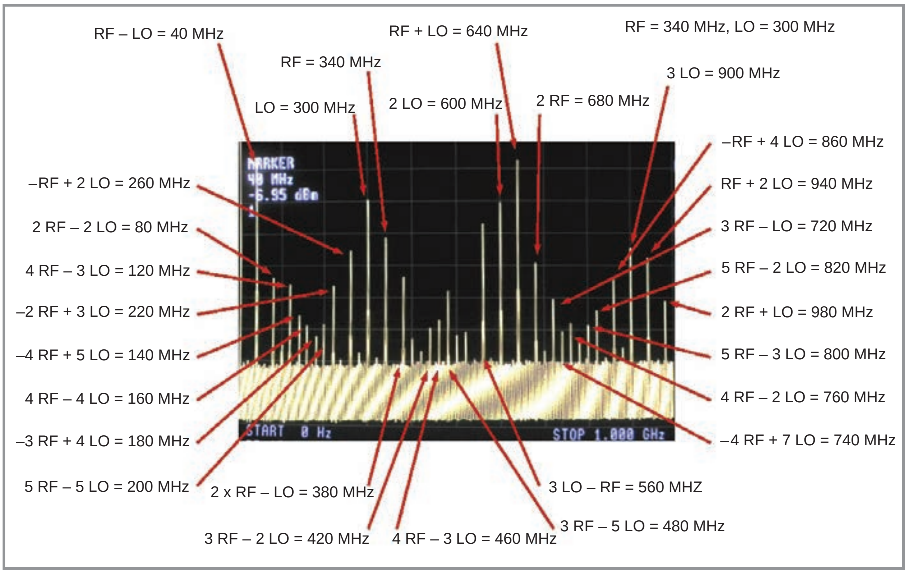



**Figure 5**—Starting with the spectrum shown in Photo 3c, it is interesting to check the origin of each frequency. I did it for you.

A last word on conversion losses: Remember my 2011 article "Noise Figures 101" (*Circuit Cellar* 249)? The noise figure of a passive mixer is equal to its conversion loss, so the lower the better.

What else? You can also check the mixer isolation figures. As explained earlier, this is a measure of the leakage between ports from LO to RF and from LO to IF, but also from RF to IF. This isolation is linked to the mixer topology (i.e., single-balanced versus double-balanced or even triple-balanced configuration), but also to the quality and matching of the internal components. Better mixer balance translates to lower leakage. You can usually find isolation figures of 20 to 30 dB, which is not high in particular as the mixer losses are already significant. Why? Let's assume that the RF to IF isolation is 20 dB and the conversion loss is 9 dB. Then there will be just 11 dB difference between the IF signal desired and the RF input signal leakage to the IF output port, which is not much.

Lastly P1dB, which stands for 1 dB compression point, is the input power for which the conversion loss of the mixer is 1 dB greater than its value at low power levels. So, P1dB corresponds to a point where the mixer is "compressing" the input power due to its own saturation, at the specified LO power. However, it is unusual to come close to the P1dB power, as second- and third-order distortions are usually a problem for far lower powers. The closer to the compression point, the higher the distortion. You will see significant distortion 10 dB or even 20 dB lower than the P1dB specified power. I can't go into more detail this time, but if you are interested, Mini-Circuits's application note "Understanding Mixers – Terms Defined, and Measuring Performance" provides a good introduction to nonlinearity and intermodulation mixer parameters.

## SOME ADVICE

Let's summarize what you should do when playing with a mixer. Never use it close to its maximum input power. Select the LO power orders of magnitude higher than your highest input signal. Think twice about the possible spurious frequencies and select a good set of RF/LO/IF frequencies to avoid the nastiest ones. And remember that a mixer must always be used with a filter to select the signal in which you are interested.

By the way, there is another useful trick here: you will often see that RF designers like to add a passive attenuator pad between the mixer output and the filter. Why? A mixer already has a huge conversion loss, why increase it with an attenuator? The reason is to get a cleaner output. For example, an LC mixer has basically no dissipating components. The signals filtered out are not transmitted to the output, but they are reflected back to the mixer as they can't be dissipated in the filter. Due to the nonlinear components inside the mixer, these reflected signals, coming from its output, could be mixed again with the RF and LO, increasing the number of spurious signals. The attenuator reduces the useful signal, but reduces twice the reflected signal so it attenuates the problem. This is not always required, but it could help.

## WRAPPING UP

As usual, I feel like I have only covered 10% of the subject, and I probably have not been precise enough for even that 10%. I did not even mention the useful mixer variants, such as image-reject mixers, harmonic mixers, or the IQ mixers now used everywhere. The good news is that I have subjects ready for my future articles. Anyway, I hope you liked this quick overview. Don't hesitate to play with mixers; they are fun and not too complex as long as you understand how they work. Even if you don't have RF generators and a RF spectrum analyzer, you can build some interesting experiments at audio frequencies, perhaps using your PC sound card as a spectrum analyzer? Mixers are working as soon as you are no longer at 0 Hz. In my next article, I will continue to discuss mixers, showing you some tools that can make your life easier when designing an RF chain. ■

*Robert Lacoste lives near Paris, France. He has 22 years of experience working on embedded systems, analog designs, and wireless telecommunications. He has won prizes in more than 15 international design contests. In 2003, Robert started a consulting company, ALCIOM, to share his passion for innovative mixed-signal designs. You can reach him at rlacoste@alciom.com. Don't forget to write "Darker Side" in the subject line to bypass his spam filters.*

## PROJECT FILES

To download the code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2012/263.

## RESOURCES

Hittite Microwave Corp., Mixer Spur Chart Calculator, www.hittite.com/tools/Spurcalc/SpurCalc25JarLockE.htm.

R. Lacoste, "Noise Figures 101," *Circuit Cellar* 249, 2011.

Marki Microwave, Inc., "Mixer Basics Primer: A Tutorial for RF & Microwave Mixers," 2012, www.markimicrowave.com/menus/appnotes/mixer_basics_primer.pdf.

Mini-Circuits, "Understanding Mixers – Terms Defined, and Measuring Performance," 2008, www.minicircuits.com/app/AN00-009.pdf.

Radio-Electronics.com, "RF Mixer and RF Mixing Tutorial," www.radio-electronics.com/info/rf-technology-design/mixers/rf-mixers-mixing-basics-tutorial.php.

I. Rosu, YO3DAC - VA3IUL, "RF Mixers," www.qsl.net/va3iul/RF%20Mixers/RF_Mixers.pdf.

## SOURCES

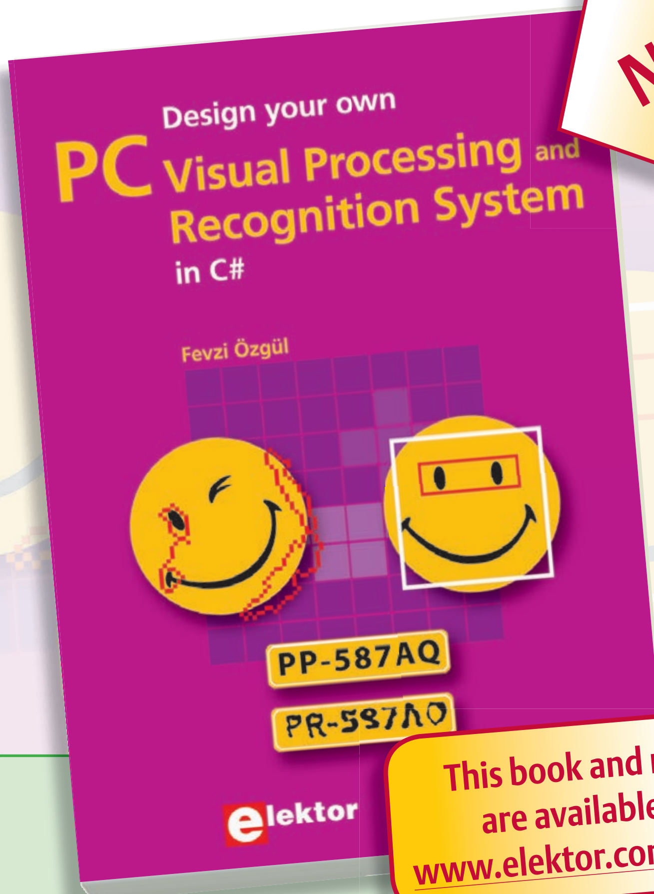
**71200C Microwave spectrum analyzer (discontinued)**
Agilent Technologies | www.agilent.com

**Proteus Isis CAD tool suite and simulator**
Labcenter Electronics Ltd. | www.labcenter.com

**MD-525-4 Double-balanced mixer**
M/A-COM Technology Solutions Holdings, Inc. | www.macomtech.com

# Elektor Shop

## The world of electronics at your fingertips!




NEW!

This book and more are available at www.elektor.com/books

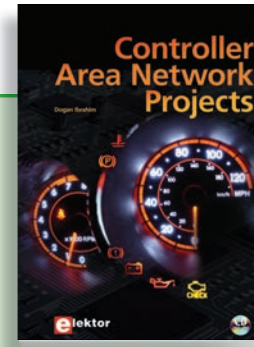
Books

A comprehensive and practical how-to guide

### Design your own PC Visual Processing and Recognition System in C#

This book is aimed at Engineers, Scientists and enthusiasts with developed programming skills or with a strong interest in image processing technology on a PC. Written using Microsoft C# and utilizing object-oriented practices, this book is a comprehensive and practical how-to guide. The key focus is on modern image processing techniques with useful and practical application examples to produce high-quality image processing software. Analysis starts with a detailed review of the fundamentals of image processing. It progresses to explain and explore the practical uses of two highly sophisticated and freely downloadable, open source image processing libraries; AForge.NET and Emgu.CV, utilizing dotnet technology within the Microsoft Visual Studio environment. All code examples used are available – free of charge – from the Elektor website; you can easily create and develop your own results to demonstrate the concepts and techniques explained.

307 pages • ISBN 978-1-907920-09-7 • $57.30

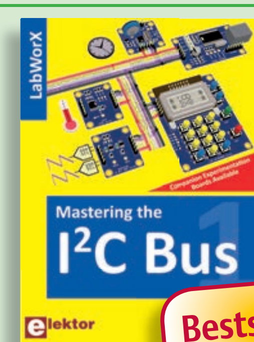



Free mikroC compiler CD-ROM included

### Controller Area Network Projects

The aim of the book is to teach you the basic principles of CAN networks and in addition the development of microcontroller based projects using the CAN bus. You will learn how to design microcontroller based CAN bus nodes, build a CAN bus, develop high-level programs, and then exchange data in real-time over the bus. You will also learn how to build microcontroller hardware and interface it to LEDs, LCDs, and A/D converters. This book comes complete with a free CD-ROM.

260 pages • ISBN 978-1-907920-04-2 • $47.60




Bestseller!

LabWorX: Straight from the Lab to your Brain

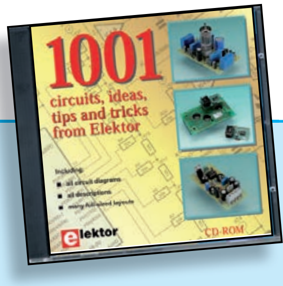
### Mastering the I²C Bus

Mastering the I²C Bus is the first book in the LabWorX collection. It takes you on an exploratory journey of the I²C Bus and its applications. Besides the Bus protocol plenty of attention is given to the practical applications and designing a solid system. The most common I²C compatible chip classes are covered in detail. Two experimentation boards are available that allow for rapid prototype development. These are completed by a USB to I²C probe and a software framework to control I²C devices from your computer.

248 pages • ISBN 978-0-905705-98-9 • $47.60

Prices and item descriptions subject to change. E. & O.E

# CD/DVD-ROMs




**Circuits, ideas, tips and tricks from Elektor**

## CD 1001 Circuits

This CD-ROM contains more than 1000 circuits, ideas, tips and tricks from the Summer Circuits issues 2001-2010 of Elektor, supplemented with various other small projects, including all circuit diagrams, descriptions, component lists and full-sized layouts. The articles are grouped alphabetically in nine different sections: audio & video, computer & microcontroller, hobby & model-ling, home & garden, high frequency, power supply, robotics, test & measurement and of course a section miscellaneous for everything that didn't fit in one of the other sections. Texts and component lists may be searched with the search function of Adobe Reader.

ISBN 978-1-907920-06-6 • $55.70


**A whole year of Elektor magazine onto a single disk**

## DVD Elektor 2011

The year volume DVD/CD-ROMs are among the most popular items in Elektor's product range. This DVD-ROM contains all editorial articles published in Volume 2011 of the English, American, Spanish, Dutch, French and German editions of Elektor. Using the supplied Adobe Reader program, articles are presented in the same layout as originally found in the magazine. An extensive search machine is available to locate keywords in any article. With this DVD you can also produce hard copy of PCB layouts at printer resolution, adapt PCB layouts using your favourite graphics program, zoom in/out on selected PCB areas and export circuit diagrams and illustrations to other programs.

ISBN 978-90-5381-276-1 • $37.90


## AVR Software Defined Radio

This package consists of the three boards associated with the AVR Software Defined Radio articles series in Elektor, which is built around practical experiments. The first board, which includes an ATtiny2313, a 20 MHz oscillator and an R-2R DAC, will be used to make a signal generator. The second board will fish signals out of the ether. It contains all the hardware needed to make a digital software-defined radio (SDR), with an RS-232 interface, an LCD panel, and a 20 MHz VCXO (voltage-controlled crystal oscillator), which can be locked to a reference signal. The third board provides an active ferrite antenna.

*Signal Generator + Universal Receiver + Active Antenna: PCBs and all components + USB-FT232R breakout-board*

Art.# 100182-72 • $133.00

**Elektor is more than just your favorite electronics magazine. It's your one-stop shop for Elektor Books, CDs, DVDs, Kits & Modules and much more!**

**www.elektor.com/shop**

Elektor USA
4 Park Street
Vernon, CT 06066
USA
Phone: 860-875-2199
Fax: 860-871-0411
E-mail: order@elektor.com

# Kits & Modules


## Improved Radiation Meter

This device can be used with different sensors to measure gamma and alpha radiation. It is particularly suitable for long-term measurements and for examining weakly radioactive samples. The photodiode has a smaller sensitive area than a Geiger-Müller tube and so has a lower background count rate, which in turn means that the radiation from a small sample is easier to detect against the background. A further advantage of a semiconductor sensor is that is offers the possibility of measuring the energy of each particle, allowing a more detailed investigation of the characteristics of a sample. The optional PC-based software displays the energy spectrum, permitting a very detailed analysis to be carried out.

*Kit of parts incl. display and programmed controller*

Art.# 110538-71 • $57.30

## AndroPod

With their high-resolution touchscreens, ample computing power, WLAN support and telephone functions, Android smartphones and tablets are ideal for use as control centres in your own projects. However, up to now it has been rather difficult to connect them to external circuitry. Our AndroPod interface board, which adds a serial TTL port and an RS485 port to the picture, changes this situation.

*Andropod module with RS485 Extension*

Art.# 110405-91 • $74.00

by Jeff Bachiochi (USA)

# Build an MCU-Based Automatic Blood Pressure Cuff

Personal health products are becoming more and more commonplace. They reinforce regular visits to personal physicians and can be beneficial when diagnosing health issues. This article shows you how to convert a manual blood pressure cuff into an automatic cuff by adding an air pump, a solenoid release valve, and a pressure sensor to a microcontroller circuit.

Modern electronics and embedded technologies are constantly changing the healthcare industry for the better. As engineers and electronics innovators, we're in the position to test new ideas and even bring new healthcare-related products to the market. This month I'll detail how an experience at a blood donation center led me to build my own microcontroller-based automatic blood pressure cuff (see Photo 1).

## BLOOD PRESSURE TESTING

Last year around this time, I went to a local high school during its sponsored blood drive. Interestingly, a "bloodmobile"—which is a mobile blood donation center—was set up to handle four donors at a time. Donating has gotten rather uneventful over the years, but that day changed my outlook forever.

Typically, during the pre-donation interview, a registered nurse (RN) tests your pulse, blood pressure, and iron content (with a finger prick that for some reason always hurts more than the donation itself). If all is well, you must answer a battery of questions to ascertain whether your blood might be contaminated in any way. Well, when I visited the bloodmobile, I didn't have to answer any questions. I was rejected for having high blood pressure.

On my way back home, many things were racing through my mind. It was clear I wasn't going to get any work done for the rest of the day. Instead, I Googled the potential causes of high blood pressure and found: poor eating habits, smoking, physical inactivity, stress, and more. Since I don't smoke, and I enjoy outdoor activities like running (at least during the nicer weather), it seemed like taking a good look at my diet could in fact address two of the causes with one blow. I have a penchant for fried foods and red meat. And, after a few months of decreased activity, I was tipping the scale at more than 200 lbs. It was time to change my eating habits.

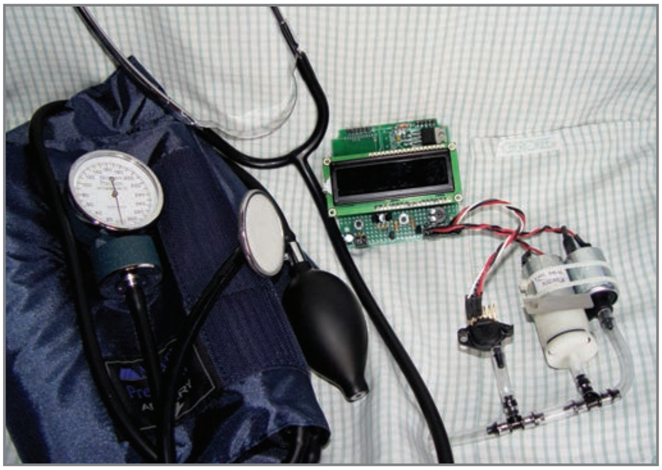


Photo 1—I turned a manual blood pressure cuff into an automatic cuff by adding an air pump, a solenoid release valve, a pressure sensor, and a microcontroller.

| Category | Systolic (mmHg) | Diastolic (mmHg) |
|---|---|---|
| Hypertension | < 90 | < 60 |
| Desirable | 90 – 119 | 60 – 79 |
| Prehypertension | 120 – 139 | or 80 – 89 |
| Stage 1 hypertension | 140 – 159 | or 90 – 99 |
| Stage 2 hypertension | 160 – 179 | or 100 – 119 |
| Hypertensive crisis | ≥ 180 | or ≥ 120 |

Table 1—Our blood pressure may vary from too weak to too strong. Pressures outside of the desirable range may cause adverse health risks.

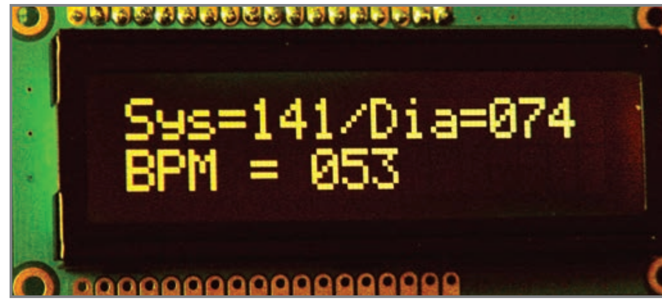



Photo 2—Test results show that during the middle of a specific day my blood pressure has a slightly elevated systolic level, as reflected in this yellow (cautionary) display color. My BPM number was a low 53.

Initially I dropped meat, grease, salt, and sugar from my meals. I lost 30 lbs in a few months and returned to donating blood successfully. Since then, I've eased off the rigorous regimen and I am attempting to find a happy medium, basically minimizing my salt and grease intake. My wife Beverly is an RN. She provided me with a blood pressure cuff (an adjustable tourniquet with a gauge that displays pressure) and a stethoscope (normally used to listen to your heart or lungs) so I could monitor my blood pressure. The idea is to apply the cuff to your upper arm. It contains a bladder that can be pumped up with air. This squeezes your arm and as the pressure rises it will eventually prevent any blood from flowing in your arm. Note: You don't want to stop the blood like this for any length of time, unless you are preventing blood loss from a severed limb.

The cuff's pressure is displayed in the range of 0–300 mm of mercury (mmHg). Placing the stethoscope on your arm just below the cuff enables you to hear the blood as it squirts past the point of compression by the cuff. Pump up the cuff's pressure until you can no longer hear the blood flowing. Slowly let out the pressure until you begin to hear the blood flowing and note the pressure (systolic). Continue

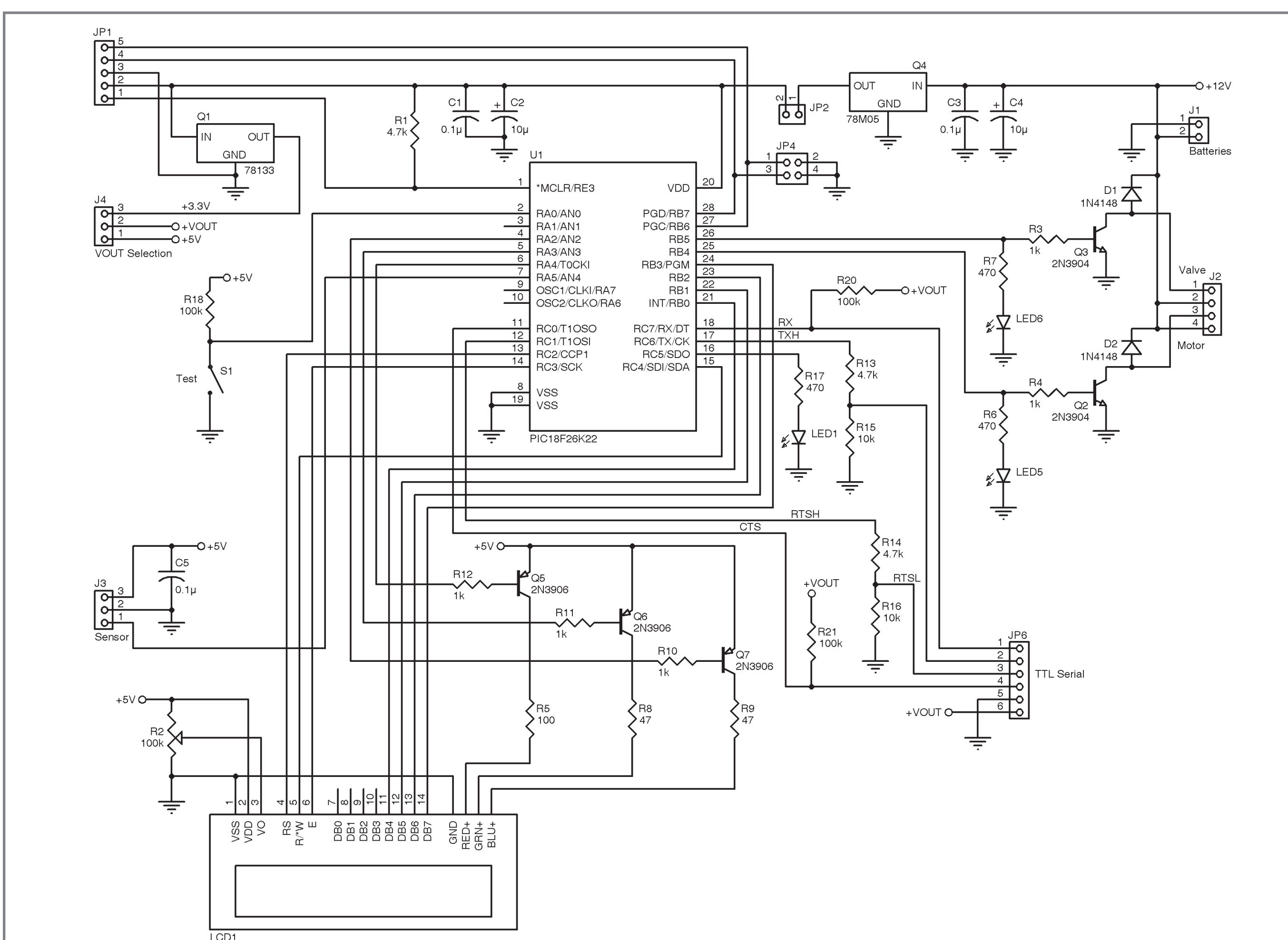



Figure 1—This is my Microchip Technology PIC18F26K22-based circuit.

to release the pressure until you can no longer hear the blood flow anymore and again note the pressure (diastolic). Your blood pressure is normally given as systolic pressure over the diastolic pressure in millimeters of mercury (e.g., 120/80). Table 1 shows blood pressure classifications for adults.

This method of measuring blood pressure (oscillometric) has been used since the late 1800s. Recently, automatic blood pressure cuffs have been on the market for around $50. I turned a manual blood pressure cuff into an automatic cuff by adding an air pump, a solenoid release valve, a pressure sensor, and a microcontroller-based circuit (see Photo 1). As is often the case, I had to set up the basic circuitry before I could begin any actual measurements. Let's review.

## BASIC CONTROL CIRCUITRY

Besides the single analog input for a pressure sensor and two digital outputs for the pump motor and solenoid valve control, I needed some kind of user interface. As a standalone project, I required a push button input to begin a test cycle and an LCD for displaying results (see Photo 2). At the start, there was plenty about this project to uncover, so I used a serial port for information and debugging purposes. I had no idea what the sampled output would look like or even what kind of algorithm I'd need to discover the diastolic and systolic pressure points and heart rate. So I needed a way to gather and look at the data. I figured a serial port would be more effective than trying to analyze stuff displayed on a small LCD.

Figure 1 depicts my circuit. Note that everything runs from 12 V (batteries). The only air pump (MPJA's 18831-MI) and solenoid valve (Jameco Electronics's TAV-0829-R) I could find at my favorite distributors required 12 VDC; otherwise, I would have preferred to use 6 V. Most of the circuitry runs on 5 V because of the LCD. I found a 2 × 16 Newhaven Display NHD-0216K1Z-NS(RGB)FBW-REV1 LCD module with a RGB backlight. I hadn't used or seen one before and was excited to try it out. A Freescale Semiconductor MPX5050 silicon pressure sensor is an integrated silicon sensor featuring on-chip signal conditioning and temperature compensation. With a 5-V input, I knew this sensor would output 0–5 for an input of 0–50 kiloPascals (kPa) or 0–375 mmHg. This covered the total range quite nicely.

The pump and valve outlets fit the 0.25″OD plastic tubing I found at a hardware store. To attach everything, I also found some "T" fittings (see Photo 1). The pump can produce up to approximately 350 mmHg at 50 mA, and the solenoid requires approximately 300 mA to hold it open. These are controlled by 2N2222 TO-92 NPN transistors.

## PRESSURE TESTING

To test the pressure sensor, I needed independent control of the air pump and the solenoid valve. To do this, I created a user interface using the serial port. I knew a number of single-character commands would enable me to do everything from a terminal program running on my PC.

```
Commands
  A(bort)
  M(otor)
  P(ressure)
  R(esults)
  T(est)
  V(alve)
  ? (list commands)

'T' when cuff is secured above the elbow
```

Figure 2—The project outputs a sign-on message plus this command menu when a terminal program opens and connects to the port. The RTS output line is asserted (TTL logic low) when the serial port connection is established.

Figure 2 shows the "sign-on message," which includes a list of the accepted commands. Single-character commands make it possible to handle all user input within the serial receive interrupt. All characters are converted to uppercase so either case is acceptable. Character recognition simply sets a corresponding bit in a request register. This register's bits are treated as if they are push-button inputs within the main application loop. Figure 3 details the main loop services.

The application places the most emphasis on sampling pressure. This is based on a recurring timer overflow. The timer merely begins a sample conversion by setting the Go bit of the ADC. Since I am dealing with a single analog channel, I don't have to worry about acquisition time, just conversion time. I initially set the timer overflow to 500 µs. That's 2,000 samples per second for testing purposes. Each 10-bit sample, measured as 0–5 V, must be converted to millimeters of mercury. This is accomplished at the end of each conversion (within the analog conversion interrupt). Each bit is equal to approximately 0.4069 mmHg. To avoid working in floating point, I use a scale factor of 256. This way each bit is multiplied by 104 (256 × 0.4069) and the result is presented in the middle byte of the 3-byte result (an easy divide by 256). A sample of 200 would convert to 81 mmHg (200 × 104/256) using floating point, this would be 81.34 (200 × 0.4069). I use an unsigned word × byte routine that is less than 20 instructions to perform the task.

Refer to Figure 2 for a sense of what is happening in this application. A Microchip Technology PIC18F26K22 microcontroller has dual-level interrupts, which means I can enable high-level interrupts to interrupt lower-level functions. There are six interrupt routines used: UART-TX, UART RC, A/D conversion complete, and three timer routines. The first timer takes care of starting A/D conversion on a regular basis. A second timer gives a 1-s tick. A third timer handles the LCD LED backlight (more on this later).

I started by building the framework for this application, with each request simply returning without doing anything. I wrote request or command routines as necessary. The first command "?" included the sign-on message that also displayed the command list for the serial port. I knew this would verify the framework is working with both an

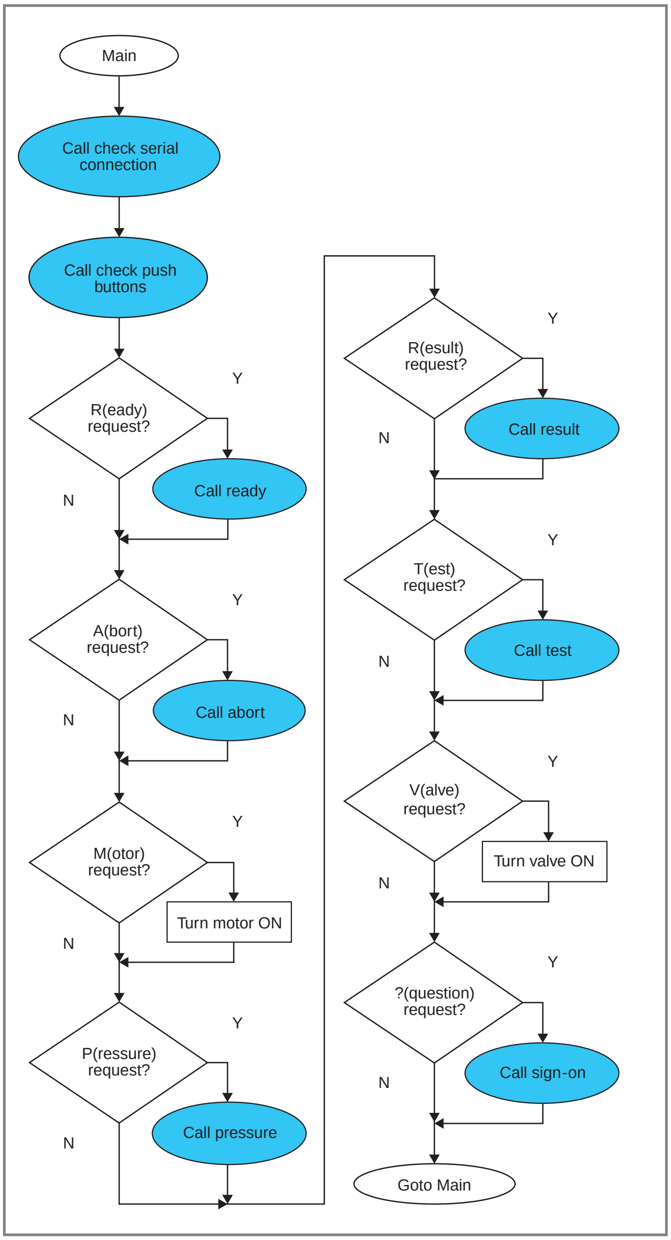



Figure 3—The Main loop services all the requests from either the physical buttons or the serial user input.

attached LCD and a UART. Next, I added M, V, and A command routines to start the motor, open the valve, and turn everything off, respectively. This gave me sufficient control over the hardware to display some pressure readings. Even at 38,400 bps, I can acquire data faster than I can send it out the UART. So, I chose to send every tenth pressure reading in a xxx<CR><LF> decimal format. I chose Tera Term as my user interface application on my PC. I can use it to copy data received into a text file.

A spreadsheet application (e.g., Excel) enables me to import and graph data easily. Figure 4 shows portions of the pressure data while inflating (motor on) and deflating (motor off). Note: There is a slow leak back through the air pump when the motor is turned off (this is actually beneficial, more shortly). I can quickly deflate the cuff by opening the solenoid valve (on). Abort turns off either the

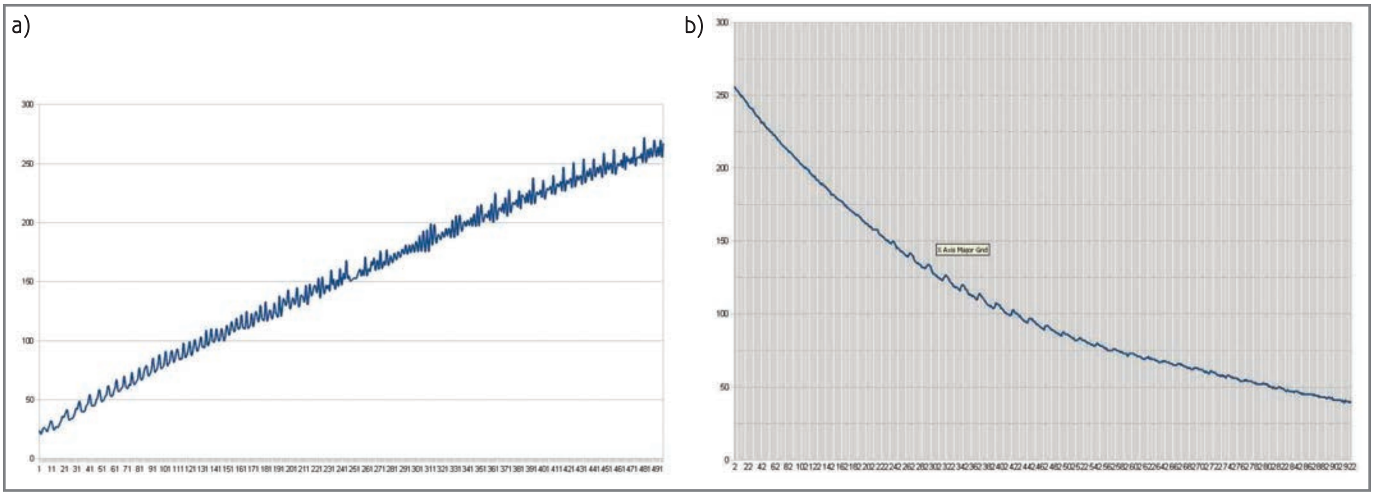



Figure 4—Portions of the pressure data while inflating with the motor on (a) and deflating with the motor off (b)

motor or the valve. The first thing you notice from the two graphs is that inflation is noisy. In fact, it looks impossible to detect any difference between pressure changes due to the rotary pumping cycle and any heartbeat pulsing. However, once off, the lack of pumping pressure changes enables blood pulsing to be seen as pressure increases while the cuff slowly deflates.

## THE DETECTION ALGORITHM

It was pretty clear that with the air pump I was using I wouldn't be able to filter out the 20 mmHg of pressure deviation I saw from the air pump and still be able to see 5 mmHg of change from the blood pulsing. So, I had to concentrate on the deflation period to obtain any information. I expected to see a range of 40 to 160 beats per minute (BPM). These numbers translated into periods of 1.5 s and 400 ms, respectively. The fastest period (400 ms) × 2 was the minimum sample rate (Nyquist) to pick out the peaks and valleys. While this would be 200 ms, I chose a faster rate so I could average out any potential noise.

I tried to work with simple byte values. If I sampled every 10 ms, I could take 255 samples before I rolled past 1 byte: that's 2.55 s (255 × 10 ms). That was plenty of time to catch the 40 BPM. At 160 BPM, that's 40 samples (400 ms/10 ms). I could have reduced the sampling frequency to 100 samples per second (1/10 ms), but I decided on an average of 16 samples to smooth out any funny stuff. The divide by 16 (nibble) was pretty simple and the conversion from this average to BPM was another calculation that used an unsigned word/byte routine, which again eliminated the need for floating point. BPM = seconds × sample rate/SampleCount or (6,000/SampleCount). I knew the slowest BPM, 40, would be 150 (6,000/40) and the fastest BPM, 160, would be approximately 37 (6,000/160). Any SampleCount less than 37 would be considered too fast and any SampleCount greater than 150 would be considered too slow.

Note: SampleCount is capped at 255 to keep it to a single byte, as it can't be any slower than too slow. Actually, a too-slow count is really useful. Got any ideas?

Once the new sampling rate was changed in the application, I ran another test. I wanted to get a better look at what I was trying to discover, the point where I could detect a rise in the deflating pressure, due to a throb of blood spurting by the cuff's constriction.

Figure 5 shows a close up of a single heartbeat. You can see the pressure falling for FallCount samples. I define FallCount as the number of consecutive samples the value drops or remains the same. Likewise, I define RiseCount as the number of consecutive samples the value rises or remains the same. Should the value change direction even once, the new direction's count is initialized to 1. At any point in time there is both a FallCount and a RiseCount. You can see that for a complete pulse the value will fall for some FallMinimum count and then it will rise for some RiseMinimum count. I define a legal cycle as at least one FallMinimum count and then one RiseMinimum count.

Now here's where the too slow count comes into play. When using my design, once deflation begins, I want to make sure that I have some period of time (greater than 150 samples, too slow) before detecting any pressure rise or I may not have inflated past the point of the systolic pressure. By rights, I would need to reinflate to a higher

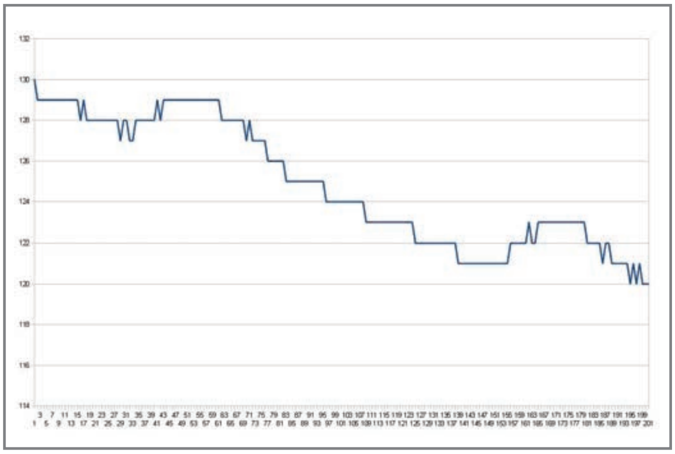


Figure 5—A single heartbeat

maximum pressure. So, the too-slow count can help determine if I've not seen a legal pulse before a too-slow count. Otherwise, the first legal pulse indicates the systolic pressure, and I save the present value (in mmHg) to the systolic register for later display. On subsequent legal pulses, the present value (in mmHg) is saved to the diastolic register for later display. This is done repeatedly until another too-slow count is found. This indicates that the heartbeat can no longer be detected and the last diastolic pressure saved is a good one.

Every time a legal pulse has been detected, the BPM is calculated. This calculation is based on the number of samples taken since the last legal pulse. Remember that a Sample Count less than 37 is considered too fast and greater than 150 is considered too slow. So, if the SampleCount is within these bounds, it is converted to BPM and saved in the BPM register for later display.

Once the pulse is lost, the test phase is just about finished. I open the valve to release the remaining pressure in the cuff until it falls to less than 20 mmHg. Then the results of the test are displayed. The results include the systolic pressure, the diastolic pressure, and the BPM count.

## NO SERIAL PORT

The serial port proved to be indispensable for analyzing pressure readings and debugging the application. However, it is not necessary for the final application. As a standalone product, a single push button and the 2 × 16 character LCD are all I need. You can see from the schematic that I am using the 4-bit mode on this display. The only difference you may have noticed is the backlight has three inputs. There is one input for each of the RGB LEDs on the side-lit backlight. Each input is controlled by a PNP transistor and an associated series resistor that sets the LED current to 20 mA. By using transistors to control the current to each LED, they can be turned on or off independently. The display (characters) can now be presented in color. Since this is a transmissive display, it must be backlit with at least one color for the displayed characters to be visible. However, by varying the on times of each color, I can get more than just RGB.

This brings me back to the purpose of the third timer in this application. It is a 100-µs timer that increments PWM Count. Three registers—RedPWM, BluePWM, and GreenPWM—can be set with a value of 0 to 100. These registers are used by Timer 3 to determine whether each LED should be off (0), on (100), or have a duty cycle (1–99) associated with it. While PWMCount is less than 100, any LED will be turned off if its pulse-width modulation (PWM) value matches the PWMCount. When PWMCount reaches 100 it is cleared. While PWMCount is greater than 0, any LED will be turned on when its PWM value = PWMCount. For instance, I find that a mix of equal parts RGB gives a white light. The PWM values used will reflect a percentage of the 20-mA current that is applied (averaged) to each colored LED. With a simple call such as "SetToWhite," the three PWM values for each color are updated with the proper mix to display the selected color.

I started out using a white backlight and changed to a color to display the results of a test. The color emphasizes the resultant value of the systolic pressure. Green indicates results in the acceptable range, yellow indicates an elevated pressure, and red indicates a result that is in the high range.

Note: This is for experimental use only and should not be used as an absolute indicator of your health. If you are concerned with having high blood pressure, please see your physician.

## WHAT NOW?

Possibly the most useful function I could add to this design would be the ability to date stamp and log data. Handling multiple users might be useful too. This would add a new level of complexity to what has been fairly simple (other than determining what algorithm to use). Definitely more than a single button will be necessary to enable a user to set the time, choose users, page through data, and so forth.

We are going to see a steady rise in personal health products that aid us in between visits to our physicians. Hardly an evening in front of the television goes by without seeing a commercial about "receiving testing supplies right to your door." Access to outpatient testing results may become paramount as an aid in a doctor's diagnosis. While this is already standard practice in the ambulatory monitoring of targeted patients, we're going to see more individuals taking a more active role in monitoring themselves in an attempt to take charge of their lives. ■

*Jeff Bachiochi (pronounced BAH-key-AH-key) has been writing for* Circuit Cellar *since 1988. His background includes product design and manufacturing. You can reach him at jeff.bachiochi@imaginethatnow.com or at www.imaginethatnow.com.*

## RESOURCE

WebMD, "Hypertension/High Blood Pressure Health Center: Causes of High Blood Pressure," www.webmd.com/hypertension-high-blood-pressure/guide/blood-pressure-causes.

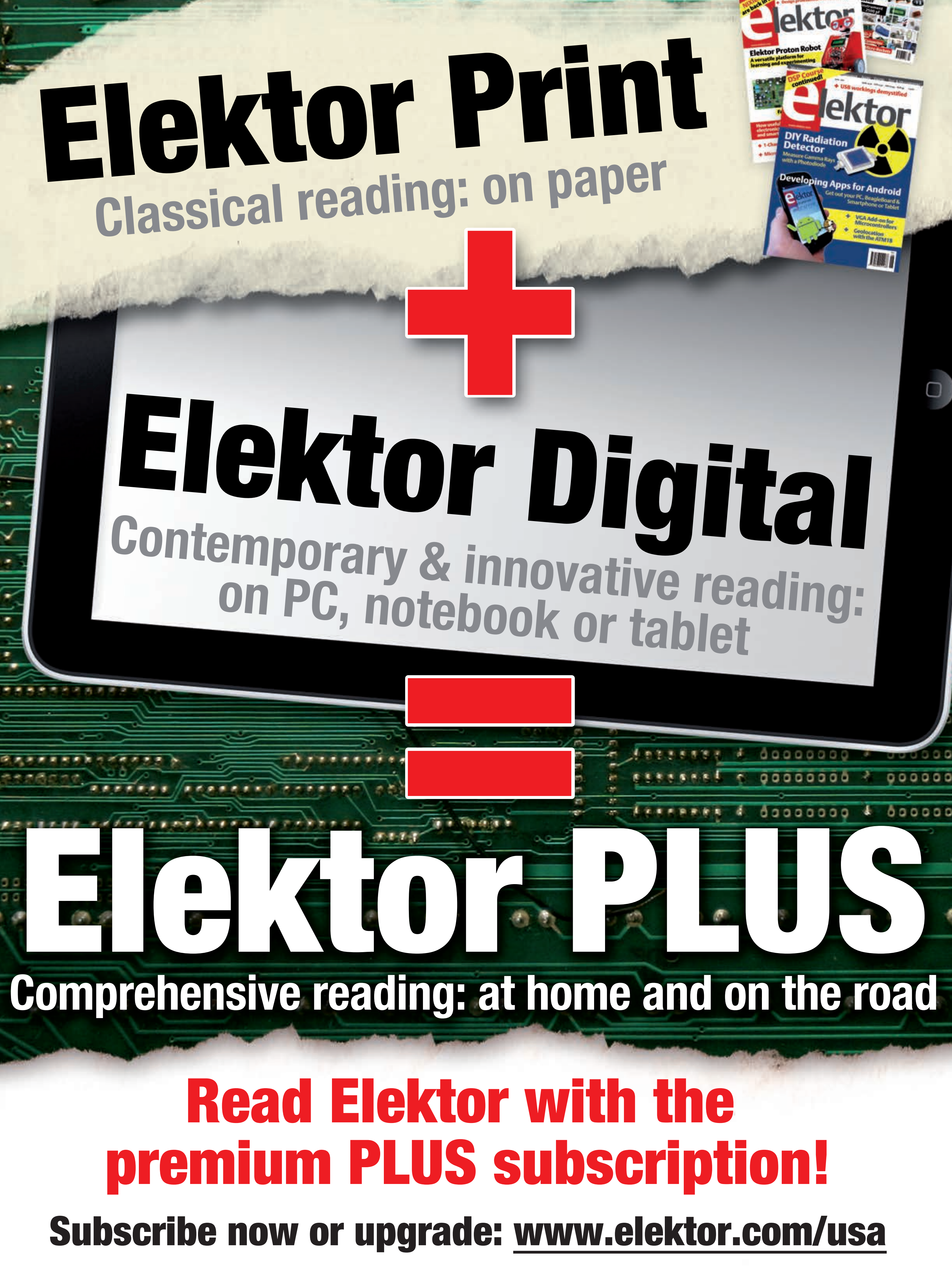
## SOURCES

**TAV-0829-R Solenoid valve**
Jameco Electronics | www.jameco.com

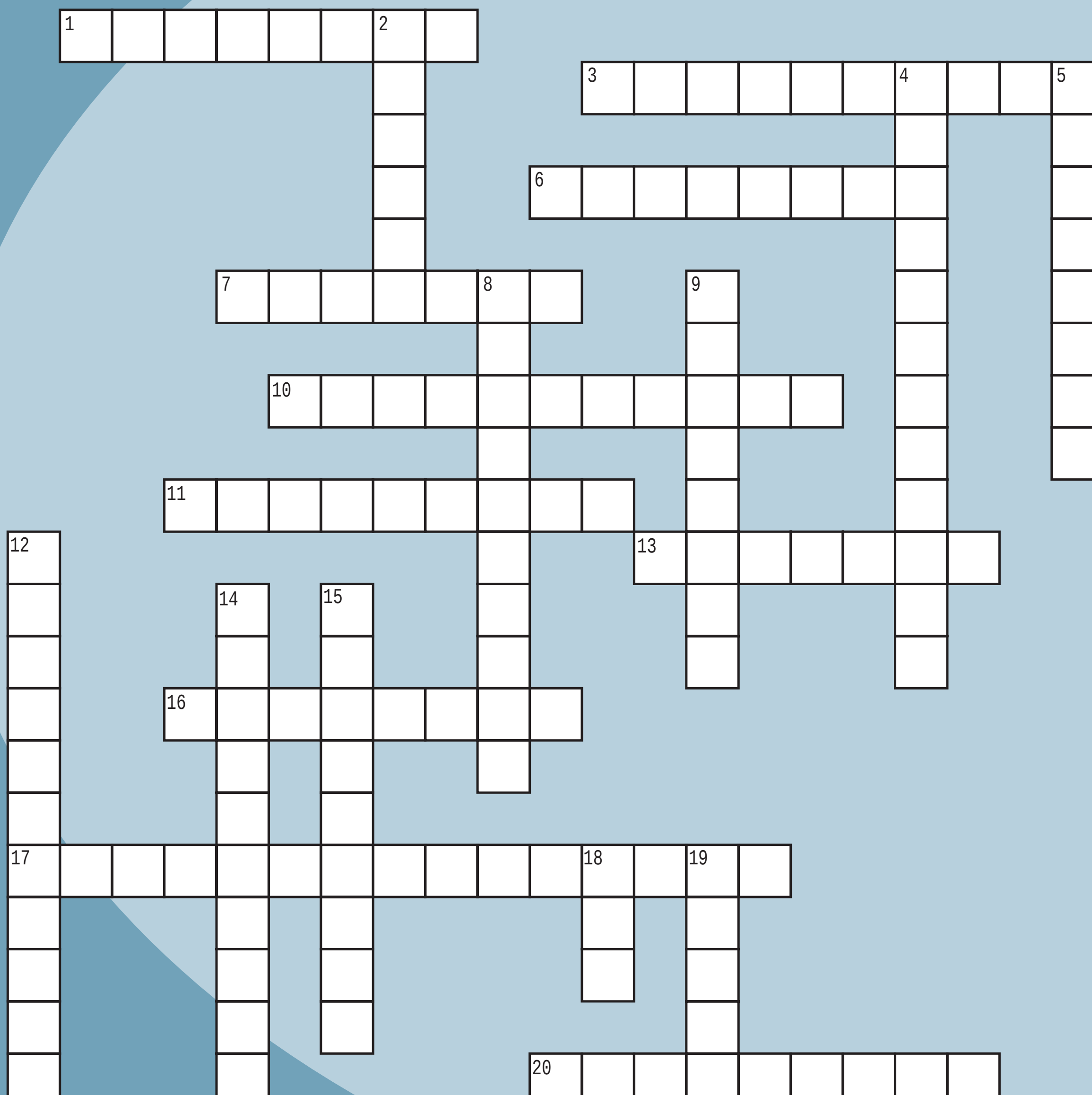
**PIC18F26K22 Microcontroller**
Microchip Technology, Inc. | www.microchip.com

**Newhaven Display NHD-0216K1Z-NS(RGB)FBW-REV1 LCD and Freescale Semiconductor MPX5050 Silicon pressure sensor**
Mouser Electronics (distributor) | www.mouser.com

**18831-MI Mini air pump**
MPJA, Inc. | www.mpja.com


Elektor Print
Classical reading: on paper
elektor
Elektor Proton Robot
DSP Course continued!
DIY Radiation Detector
Developing Apps for Android
+
Elektor Digital
Contemporary & innovative reading: on PC, notebook or tablet
=
Elektor PLUS
Comprehensive reading: at home and on the road
Read Elektor with the premium PLUS subscription!
Subscribe now or upgrade: www.elektor.com/usa

# CROSSWORD



## Across

1. A space in an otherwise closed electric circuit [two words]
3. Began in 1965 with Lotfi Zadeh's proposal of a certain type of set theory [two words]
6. Cancels a device's signal input by negative feedback from its output [two words]
7. One of these is equal to $6.28 \times 10^{18}$ electrons
10. Foucault currents [two words]
11. German physicist (1824–1887) who created a law of thermochemistry
13. Relies on truth and logic [two words]
16. Invented the LED while working at General Electric in the 1960s
17. Doesn't rely on movement or contact to switch electric circuits [three words]
20. A circuit with a current that goes one way and then another [two words]

## Down

2. Run hot and cold
4. Changes electrical signals to light and back into electrical signals
5. Adds a bit to make things even or odd [two words]
8. 1,000,000 pF
9. French engineer (1857–1926); augmented Ohm's law
12. Temperature scale where 0° = absolute zero [two words]
14. Turns light into current or voltage
15. A small tube used at very high frequencies [two words]
18. Causes 1 cm of movement
19. A character-encoding system used to represent text

**The answers will be available in the next issue.**

# IDEA BOX

## THE DIRECTORY OF PRODUCTS AND SERVICES

**AD FORMAT:** Advertisers must furnish digital files that meet our specifications (www.circuitcellar.com/advertise). ALL TEXT AND OTHER ELEMENTS MUST FIT WITHIN A 2" x 3" FORMAT. E-mail adcopy@circuitcellar.com with your file or send it to IDEA BOX, Circuit Cellar, 4 Park Street, Vernon, CT 06066. For current rates, deadlines, and more information contact Peter Wostrel at 978.281.7708 or peter@smmarketing.us.

The Vendor Directory at www.circuitcellar.com/vendor is your guide to a variety of engineering products and services.

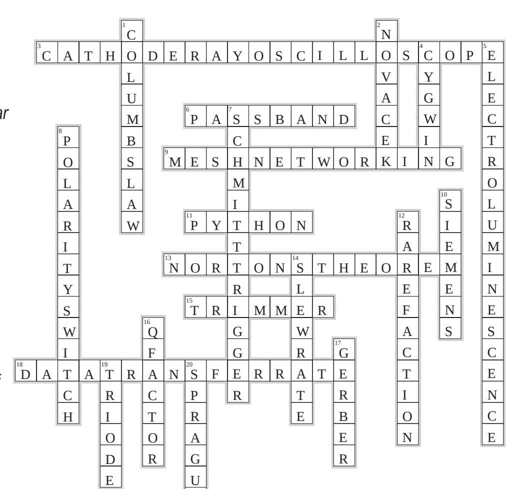
## CROSSWORD ANSWERS from Issue 262

**Across**

3. **CATHODERAYOSCILLOSCOPE**—Isn't a digital 'scope [three words]
6. **PASSBAND**—A super-efficient band of frequencies
9. **MESHNETWORKING**—Relies on nodes to capture, distribute, and reproduce data [two words]
11. **PYTHON**—An open-source programming language that relies on automatic memory management
13. **NORTONSTHEOREM**—A formula for linear electrical networks [two words]
15. **TRIMMER**—Variable resistors, variable capacitors, or trimmable inductors, for example
18. **DATATRANSFERRATE**—Measures data's speed of travel [three words]

**Down**

1. **COLUMBSLAW**—Electrostatic physics principal [two words]
2. **NOVACEK**—Wrote 26 feature articles for *Circuit Cellar* between 1999 and 2004
4. **CYGWIN**—Enables you to "get that Linux feeling on Windows"
5. **ELECTROLUMINESCENCE**—Light emission caused by electrical influences
7. **SCHMITTTRIGGER**—A comparator with two different threshold voltage levels [two words]
8. **POLARITYSWITCH**—Reverses an output signal's absolute phase [two words]
10. **SIEMENS**—Equivalent of amperes per volt
12. **RAREFACTION**—Sound wave phase
14. **SLEWRATE**—Represents a signal's maximum rate of change [two words]
16. **QFACTOR**—Measures the damping of resonator modes
17. **GERBER**—PCB file format
19. **TRIODE**—An amp device with three electrodes
20. **SPRAGUE**—Known as the "father of electric traction"



# PRIORITY INTERRUPT

by Steve Ciarcia, Founder and Editorial Director

## Google Lunar X PRIZE

This is certainly an exciting time to be an engineer. We have seen the success NASA has had with robotic exploration, especially on nearby planets such as Mars. Contrary to everything coming from NASA in the future, however, thanks to the advances in robotics and launch vehicles, "space" will soon become the province of private enterprise and not just government. Very soon, commercial space flight will become a reality.

The Google Lunar X PRIZE (www.googlelunarxprize.org) provides a focal point for these efforts. Google is offering a $20 million prize to the first team to complete a robotic mission to the moon. The basic goal is to put a lander on the surface of the moon, have it travel at least 500 m once it's there, and send back high-definition pictures and video of what it finds. There's a $5 million second prize, and also $5 million in bonus prizes for completing additional tasks such as landing near the site of a previous NASA mission, discovering water ice, traveling more than 5,000 m while on the surface, or surviving the 328-hour lunar night.

When the Lunar X PRIZE registration closed in December 2010, a global assortment of 33 separate teams had registered to compete. Seven of those teams have subsequently dropped out, but there are still 26 active teams, including 11 from the U.S. The first launch is expected sometime in 2013, and there's plenty of time before the competition ends December 31, 2015. Some teams are even planning multiple launches to improve their chances of winning.

It's interesting to browse through the team information and see the vast diversity in the approaches they're taking. This is the part that is most exciting from an engineering point of view. Some teams are building their own launch systems, while others are planning to contract with existing government or commercial services, such as SpaceX. There's a huge amount of variety among the landers, too: some will roll, some will walk, and some will fly across the moon in order to cover the required distance. Each one takes a different approach to dealing with the difficult terrain on the moon, and issues such as the raw temperature extremes between blazing sunlight and black space.

This sort of diversity is a powerful driver for future development. Each approach will have its strengths and weaknesses, and there will certainly be some spectacular failures. Subsequent missions will draw on the successful parts of each prior one. Contrast this to the approach NASA has tended to take of putting all its effort into a single design that had to succeed.

It's also interesting to consider the economics of this sort of competition. The prize doesn't really approach the full investment required to succeed. Indeed, Google is quite up front about the fact that it probably only covers about 40%, based on other recent high-tech competitions such as the Defense Advanced Research Projects Agency's DARPA Grand Challenge and the Ansari X PRIZE. This means the teams need to raise most of their money in the private sector, which keeps them focused on technologies that are commercially viable.

I have long been a fan of "hard" science fiction, as typified by writers such as Larry Niven, Arthur C. Clarke, and Michael Crichton. To me, hard science fiction means you posit a minimal set of necessary technologies, such as faster than light (FTL) space travel or self-aware computers/robots, and then explore the implications of that universe without introducing new "magic" whenever your story gets stuck. In particular, Larry Niven's "Known Space" universe—particularly in the near future—includes extensive exploration of the solar system by private entrepreneurs. With the type of competition fostered by the Google Lunar X PRIZE, I see those days as being just around the corner.

The competition among these teams, and the commercial companies that arise from them, will be good for society as a whole. For one thing, we'll finally see the true cost of getting to space, as opposed to the massive amounts of money we've been pouring into NASA to achieve its goals. As a public agency, NASA has many operational constraints, and as a result, it tends to be ultra-conservative in terms of risk taking. Policies that dictate incorporating backups for the backups certainly makes a space mission more expensive than the alternative.

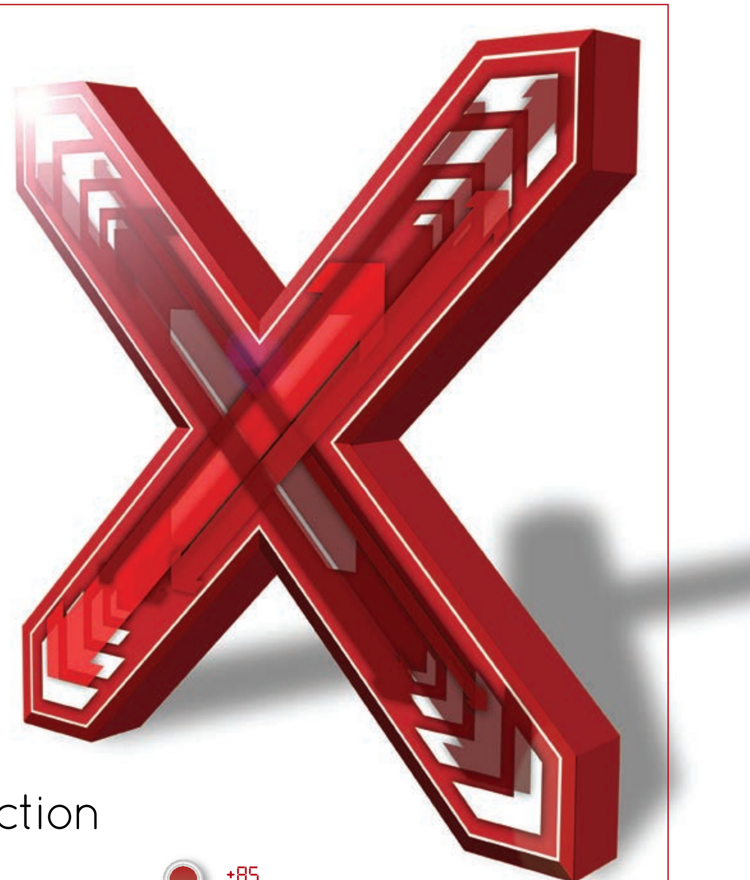
Despite these remarks, however, I don't mean to sound overly negative about NASA at all. It has had many spectacular successes, starting with the Mercury, Gemini, and Apollo manned space programs, as well as robotic exploration of the solar system with the likes of Pioneer and Voyager, and more recently with the remarkable longevity of the Mars rovers, Spirit and Opportunity. There have been many beneficial spin-offs of the space program and we have all benefitted in some way. We wouldn't be where we are today without the U.S. space program. But the future is yet to be written. There are striking differences between a publicly run space program and the emerging free-market privately funded endeavors. We would do well to recognize the opportunities and the potential benefits.

steve.ciarcia@circuitcellar.com

# FTDI Chip

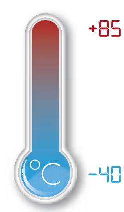
## ARE YOU READY FOR THE
# X-CHIP FACTOR?

## INTRODUCING THE X-CHIP USB BRIDGE FAMILY

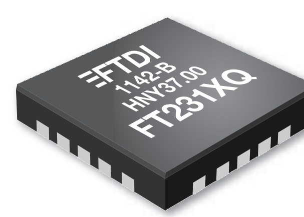


### EXtensive Interface Selection
UART, FIFO, SPI, I2C, FT1248

### EXtended Temperature Range




### Battery Charger DeteXion




### EXpandable Clocking
INTERNAL CLOCK GENERATION
EXTERNAL CLOCK OUT

### EXceptional Support

**Additional features:**

- Royalty free, validated USB driver support
  Windows, Android, Linux & Mac OS
- Low active power: 8mA
- Internal MTP memory for flexibility
- Integrated 3.3V level converter

us.sales@ftdichip.com
www.ftdichip.com